UNIVERSITY OF OSLO
Department of Informatics

Automatic segmentation of mine like objects in synthetic aperture sonar images

Gjermund Stensrud

Autumn 2013

Contents

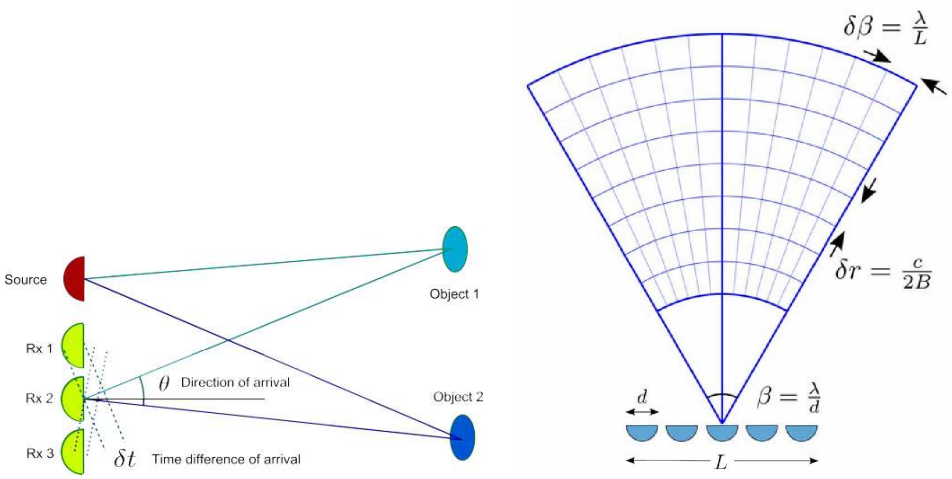
Chapter 1

Introduction

In marine warfare mining is an effective way of damaging enemy ships or preventing them from entering specific areas. Modern mines are typically placed on the ocean floor or anchored to the bottom with a chain. They activate when on physical impact or when ships are detected nearby by their noise or vibration signature. Mine hunting or Mine Counter Measures (MCM) is a way to find and destroy the mines without putting ships or people in danger. Mine sweeping is a way of removing mines that has been used for many years. It works by using ships pulling something that simulates the signature of big ships to set off the mines, a job that is slow and could be quite dangerous. In recent years Autonomous Underwater Vehicles (AUVs) equipped with sonars to scan the ocean floor has become a great new way of searching for mines. The AUVs are small and quiet which means they will not activate the mines making them well suited for the job. Newer AUVs equipped with a Synthetic Aperture Sonar (SAS) can produce high resolution images, good enough to classify objects lying on the sea floor. The task of detecting and classifying objects in the images is often done manually by trained operators looking at them with the naked eye, a process that is slow and tedious, which is why automatic detection by image analysis has become very relevant. The Norwegian Defense Research Establishment (FFI) has developed an AUV that is used, among other things, for mine hunting. Mines are typically small objects with specific shapes and can be recognized using sonar imaging and image analysis.

The NATO STO Center for Maritime Research and Experimentation (CMRE) in Italy regularly runs missions with an AUV called MUSCLE built by Bluefin Robotics Corp. and Thales. All images in this dataset is a result of missions run with MUSCLE in 2008, 2009 and 2011.

Before an object in a sonar image can be classified the object's highlight and shadow (see Figure 1.2a) needs to be identified and separated from the background, this process is called segmentation. This study will specifically focus on this segmentation process, trying out different methods and techniques and evaluate how well they perform.



(a) Direction of arrival from multiple reflectors [7]

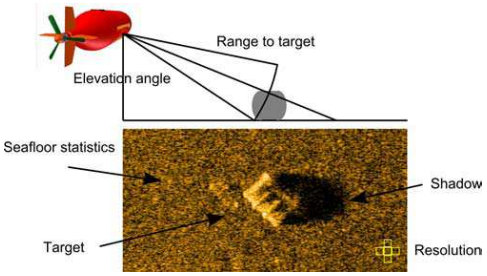
(b) Imaging sonar feld of view and resolution [7]

Figure 1.1

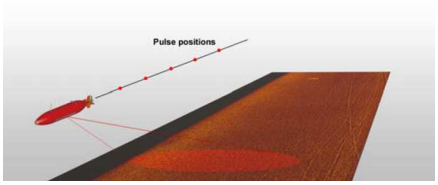
1.1 Sonar

SOund Navigation And Ranging (SONAR) has been used for ages by animals such as bats and dolphins to examine their surroundings and navigate in places where visible light is absent like in caves, or badly diffused like in muddy water. An active sonar transmits a series of sound pulses or waves and then determines distance and location of objects or obstacles by analyzing the time-delay and direction of the reflected sound. Sonar will work in any media able to propagate waves but works best in homogeneous media with a constant speed of sound. The frequency of the transmitted waves are chosen depending on the absorption properties of the media. In water the absorption rate increases with frequency which means lower frequencies will travel further than higher frequencies. The absorption also varies depending on temperature, salinity, depth and pH in addition to frequency. The range R to a reflecting object is estimated by the speed of sound c and the time delay τ: $R = \frac{c\tau}{2}$. The pulse length T_p of the transmitted signal determines the accuracy of the estimated range. The accuracy increases for shorter pulse lengths but reduces total propagation length because a shorter pulse has less energy. The accuracy is equivalent to *range resolution* δR which is defined as the minimum distance between two reflections that can be detected. Range resolution is estimated by the following formula: $\delta R = \frac{cT_p}{2}$.

The direction of the reflected signal can be estimated by using two or more receivers. The estimated direction θ is estimated from the difference in time of arrival δt between the two different receivers spaced L apart. $\theta = sin^{-1}\{\frac{c\delta t}{L}\}$. Range and direction of signals from multiple reflectors can be estimated by using multiple receivers in an array, see Figure 1.1a. By combining these techniques of estimating range and direction of multiple objects it is possible to produce an image displaying all reflected signals.



(a) Properties in a sonar image [7]



(b) Principle of synthetic aperture sonar [7]

Figure 1.2

Strong reflections give bright pixels, weak reflections give dimmer pixels and no reflection produces shadows.

1.2 Imaging sonar

An imaging sonar is a system capable of estimating range and direction of multiple objects and producing an image. The maximum range and resolution of the system is determined by the physical properties and frequency configuration of the system. The field of view β is given by the width d of a receiver element and the wavelength λ: $\beta = \frac{\lambda}{d}$. The angular resolution $\delta\beta$ is given by the array length L and the wavelength λ: $\delta\beta = \frac{\lambda}{L}$. The range resolution δR is as mentioned above: $\delta R = \frac{cT_p}{2}$, see Figure 1.1b. Note that the cross-range resolution (rotated 90 degrees from the range resolution) is dependent on the angular resolution and the range. As seen in Figure 1.1b the range lines travels further and further apart as the range increases and causes the resolution to decrease at increasing range.

When the target is an object lying on the ocean floor, imaging can be performed at any angle, like directly above, but the best results are produced when imaging from the side at a reasonable small angle above the sea floor because objects sticking out will be clearly visible by their characteristic shadow. This is called a *side-scan sonar*, see Figure 1.2a.

The biggest weakness of a side-scan sonar is that the cross-range resolution deteriorates at increasing range. A solution to this problem is to combine successive pings along a known track. This is called *Synthetic Aperture Sonar (SAS)* See figure 1.2b. This requires that the sensor platform is moved with precision along a straight path with precisely timed pings. Good images require precise navigation with good image formation and correction algorithms. This gives the need for underwater vehicles with good autonomous navigation and low self noise. This is the reason why AUVs like HUGIN and MUSCLE are built and is the best way to perform underwater sonar imaging.

To further improve the sonar images modern AUVs use a wide-band SAS to increase resolution (see Figure 1.3 for a comparison). Advanced AUVs use two receiver arrays to image the scene at a slightly different

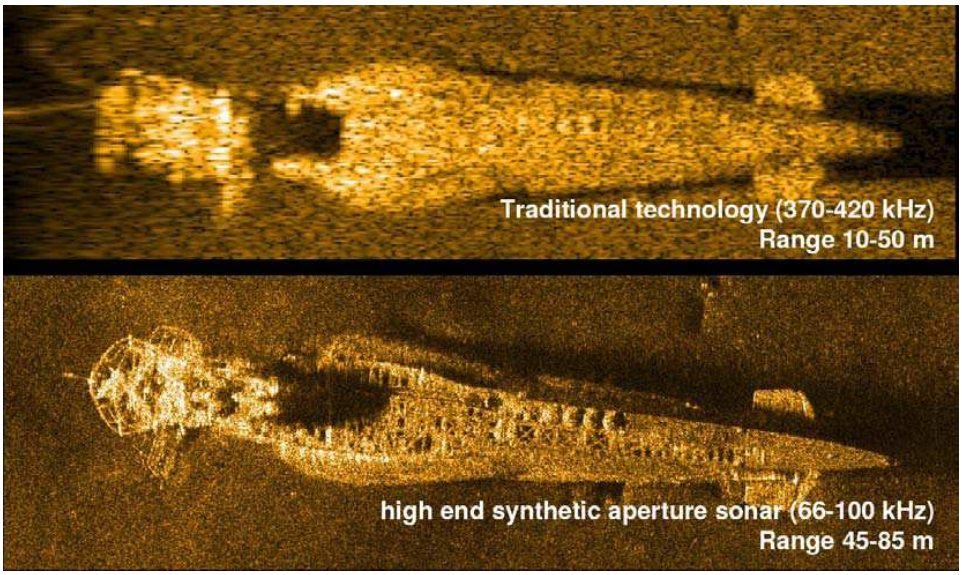


Figure 1.3: Comparison of traditional side-scan sonar with synthetic aperture sonar. Images collected by HUGIN AUV. Courtesy of Kongsberg Maritime / FFI

angle which not only increases resolution and robustness but also makes it possible to apply interferometric processing of the data which will allow 3D imaging of the scene.

A major problem with sonar images is the speckle noise, it is evenly distributed throughout the entire image. It blurs edges and details making accurate segmentation more difficult. In the past there has been done research trying to reduce the noise by using blurring filters or noise reduction filters but since the speckle is random they are only partially successful. The interesting thing about speckle noise is that it is random but reproducible. This means that if you analyze a sonar image you can find no pattern or predictability of the speckle noise, but if you image the same scene twice you will get the same speckle noise. This makes it very different from optical noise which is random, independent of scene and different every time. My first thought was that if it is reproducible it should be possible to somehow trace it back to its source, but unfortunately this is not the case. For more information on this phenomenon see chapter 4.5 of Oliver and Quegan's book about understanding SAR images [12] where they define it as multiplicative noise.

1.3 HUGIN

The HUGIN AUV, as pictured in figure 1.4, was developed in a collaborative effort by the Norwegian defense research establishment (FFI) and Kongsberg Maritime, the project includes development of both hardware and software and has been running for more than 15 years. The HUGIN system is used by the offshore survey industry for detailed seabed mapping

Figure 1.4: HUGIN 1000-MR on the Royal Norwegian Navy mine hunter KNM Hinnøy april 2008. Photo: FFI

and data acquisition and by navies for MCM (mine counter measures) and REA (Rapid Environmental Assessment). Among other sensors like multi-beam echosounder, optical camera and a sub-sea altimeter it hosts two interferometric synthetic aperture sonars (HISAS), one on each side. They operate at a typical frequency of 85 - 115kHz with along track resolution of 2.5 - 5cm and a typical range of 25-200m. HUGIN has an on-board computer performing autonomous navigation and is capable of rendering raw sonar reflection data into images.

HUGIN is capable of positioning itself and perform precise navigation and positioning based on data from internal sensors but for improved accuracy it can use a High Precision Acoustic Positioning (HiPAP) system. It is equipped with an acoustic communication link that enables the operator to be in constant communication with the AUV during the mission to monitor it's progress or execute missions in various levels of remote control. However the HUGIN AUV is able to propel itself, navigate, handle unforeseen circumstances, and achieve the mission objectives as set forth in a mission plan without human intervention.

1.4 MUSCLE

The MUSCLE AUV, as pictured in figure 1.5, developed by Bluefin Robotics Corp. with a sonar payload built by Thales is a modular vehicle specifically designed for very shallow water mine hunting. Special work has been done to reduce effects induced by shallow water such as multipath and surface reflections. Research on micronavigation as described by Bellettini and Pinto [1] has also been done to reduce errors in shallow water where surges and currents make precise navigation difficult. It has a depth rating of 80m

Figure 1.5: Muscle of length 3.5 m, which is shown deployed from the NURC coastal research vessel Leonardo in Marina di Carrara area, June 2006. [1]

with a few hours endurance and a max speed of 4.5 knots. The sonar section is a dual-side synthetic aperture sonar built by Thales Underwater Systems under CMRE specifications. A typical configuration of the sonar has a 300kHz center frequency with a 60kHz bandwidth and multiple vertical beams to reduce multipath reflections and gives a range of up to 170m at 20m depth with a range resolution of 1.5cm and 2.5cm cross-range. The on-board computer is capable of rendering the raw sonar data into sonar images using CPU and GPU processing and analyze them to assess the mission data quality. The onboard computer uses the data quality measurements for adaptive track spacing and to reacquire areas where the quality did not meet the requirements set by the operator. This level of autonomous operation saves a lot of time where otherwise subsequent missions to reacquire the same area again would be necessary.

Chapter 2

Starting point

As of today FFI have a complete working ATR processing chain that successfully performs object recognition in sonar images. The ATR process consists of four stages: Detection, Segmentation, Feature extraction and Classification. See Figure 2.1 for an overview. FFI's detection stage does a good job of processing the large sonar images looking for certain patterns or features, such as a highlight followed by a shadow suggesting potentially interesting objects, and making a list of these detections. These detections are also called regions of interest (ROIs) and are the regions analyzed further by the later stages. Their segmentation stage is based on a region growing algorithm that separates the highlight and shadow from the background and the result is then used in a template-matching classifier. This process is working well but has some shortcomings which is why they are interested in exploring other methods and algorithms in an attempt to further improve the accuracy of the system. The segmentation stage was chosen for further examination in this study because it is one of the more difficult stages and because it has a multitude of possible solutions. Segmentation is also an area of active research and new discoveries are made as more advanced methods are developed and computer hardware becomes more powerful making computationally heavy algorithms viable within a reasonable amount of time. Some of the more popular methods are described and tested below.

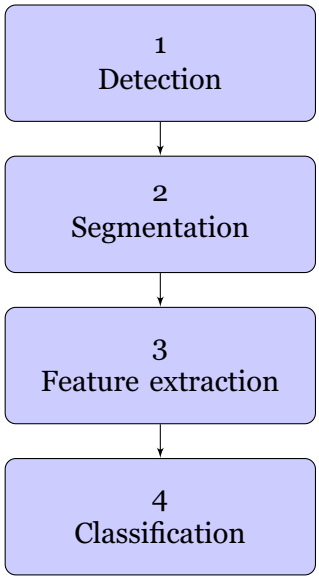


Figure 2.1: The stages of automatic target recognition (ATR)

Chapter 3

Data

The data set comes from the NATO STO Center for Maritime Research and Experimentation (CMRE) in Italy and contains MUSCLE SAS images of several different objects on different bottom types. Some example images are presented in Figures 3.1 - 3.26. Some areas contain flat uncluttered seabed where foreign objects are clearly visible (see Figure 3.3), other areas are filled with bottom vegetation, rocks or other natural shapes that can make segmentation more difficult (see Figure 3.10 and 3.16). Other areas again contain sand ripples that can confuse the object shadow and make it hard to properly segment (see Figure 3.12). Tracks made by bottom trawlers or other equipment may also confuse the object features (see Figure 3.26). None of the objects are buried under the ground or suspended above it in any way. They are all on the sea floor and should cast a shadow unless circumstances prevent it. Some of the objects have chains or wires attached to them, this is because they are test objects that are placed there and will later be retrieved after the testing is complete.

3.1 Objects

There are a few defined objects; cylinder, truncated cone and wedge. There are also some familiar objects like the oil drum and the truck wheel. The rest of the objects that we call clutter are mostly rocks, other natural shapes or different kinds of junk from ships or other vehicles. The goal is to classify the objects we are interested in without being confused by these other objects that may look similar.

The cylinder, see Figure 3.3, have a typical length of about 2m and a diameter of about 50cm. It can be found in any orientation and will look a little different from different angles, especially if it is facing away from the sensor platform, called end-shot, as in Figure 3.6 or lie parallel to the platform, called side-shot, as in Figure 3.5. It's shadow is usually some kind of skewed rectangular shape depending on orientation or in the case of an end-shot it will be cone shaped.

The truncated cone, see Figure 3.7, has a diameter at it's base at around 1m and is about 45cm tall. It will almost always be standing with it's base on the ground and look the same from every angle because of it's round

shape. The highlight will be like a half circle with a darker circle in the middle where the truncated end of the object is visible. The shadow, not surprisingly, is cone shaped and will in many cases be more recognizable than the highlight.

The wedge has a triangular shape with a blunt end and can sometimes look like a scuba fin (see Figure 3.14) because it has a narrow end that is sloping down to a wide, thin wavy edge. If the slope is facing towards the sensor platform the entire triangular top face will be visible in the highlight (see Figure 3.15), however if it is facing the other way the top face will be in shadow and give the highlight a different shape (see Figure 3.14). The shadow will have a triangular shape as well but will sometimes cover parts of the object depending on the orientation.

The oil drum, see Figure 3.22, is a standard size that is defined by Wikipedia [17] as a cylindrical shape 88cm tall with a 61cm diameter and contain 200L. Another familiar object is the truck wheel, see Figure 3.23 which is a familiar shape that needs no further description.

3.2 Challenges

Variations in bottom roughness and patterns on the ocean floor can blend in with the objects and the shadows causing problems. In Figure 3.1 a rough ocean floor causes shadows everywhere and the object shadow is mostly lost. In Figure 3.9 the rough ocean floor deforms the shadow and makes it less recognizable. Sand ripples as seen in Figure 3.24 can cause problems for both the highlight and the shadow, see Figure 3.12.

Natural objects like rocks and vegetation that might appear to have a lot of the same characteristics as the objects we are looking for is also a problem. Some rocks, like in Figure 3.17, are easier to spot while others may look very familiar, like in Figure 3.18. If the object is partially or fully concealed by other objects it will be hard or sometimes impossible to tell them apart, see Figure 3.16. If the object is in the middle of vegetation, like in Figure 3.10 and the close-up 3.11, the highlight is very hard to see but the shadow might still be recognizable.

Artifacts in Sonar imaging can cause loss of details or introduction of features where they should not be. Bright echoes can cause bright along-track lines, see Figure 3.5 and 3.6. Multipath reflections occurs when the signal bounces of several surfaces, like the ocean surface, before returning to the receiver and can cause the shadow to blend in with the background, see Figure 3.2. Vibrations or other problems at the imaging platform caused by equipment damage, ocean currents or surges can cause smudging of the image, see Figure 3.4. All these artifacts originate from the imaging system and can only be solved there. But the segmentation system needs to be aware of them and able to cope with them.

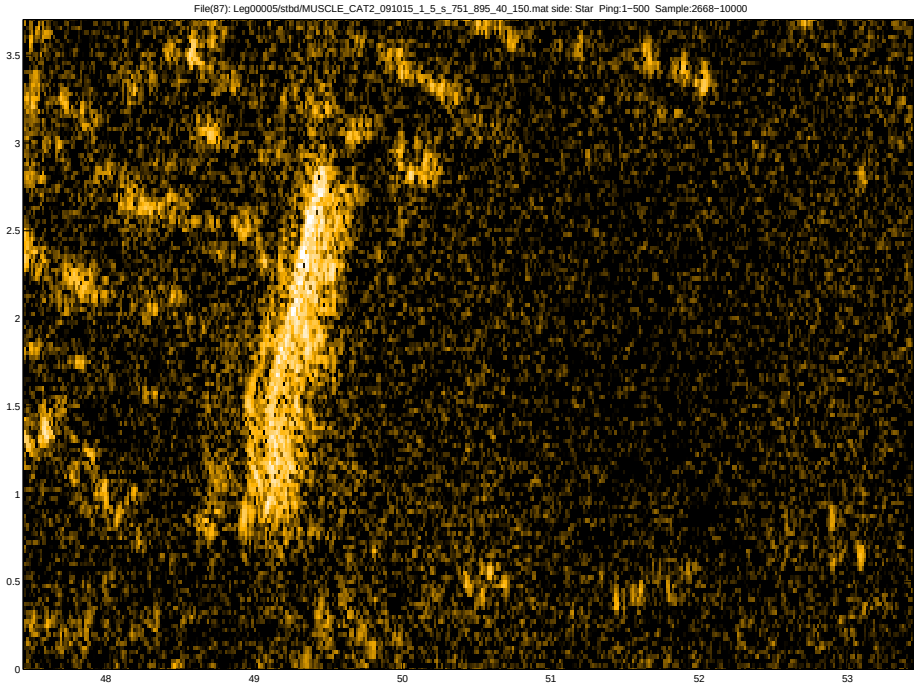


Figure 3.1: Cylinder on rough surface

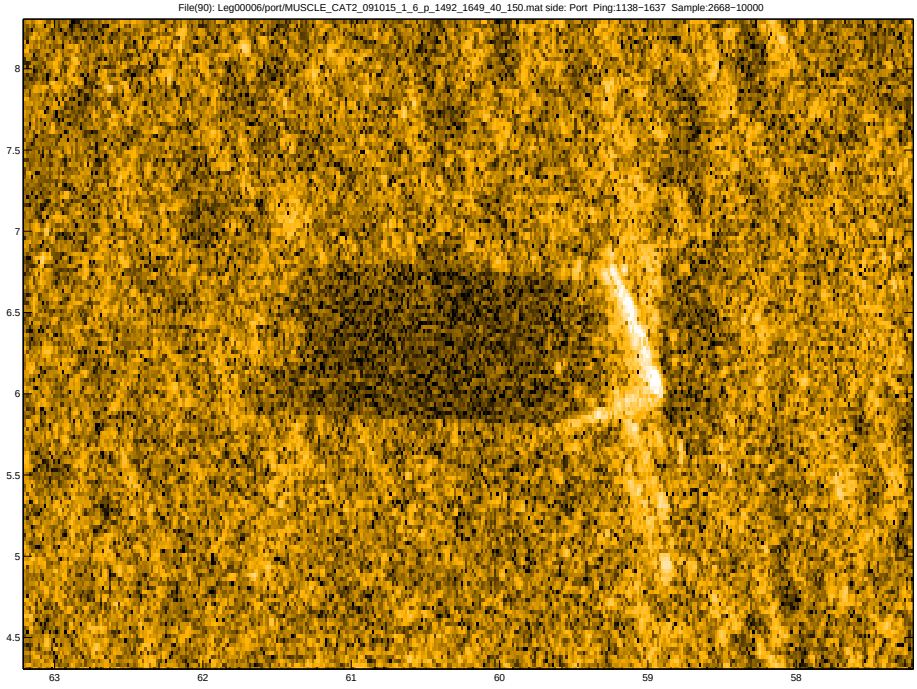


Figure 3.2: Cylinder, multipath noise in shadow

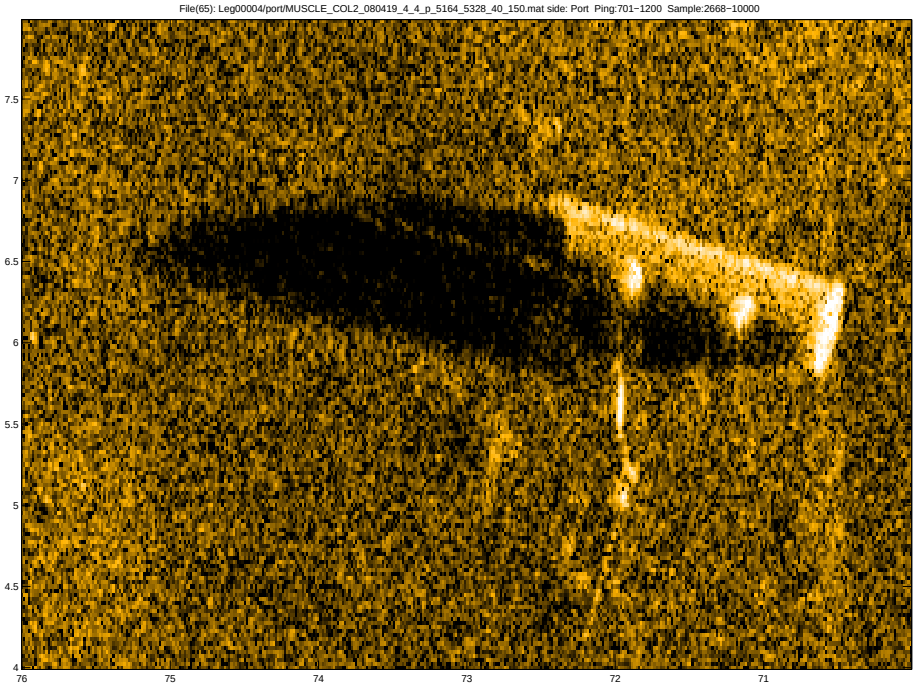


Figure 3.3: Cylinder with anchor

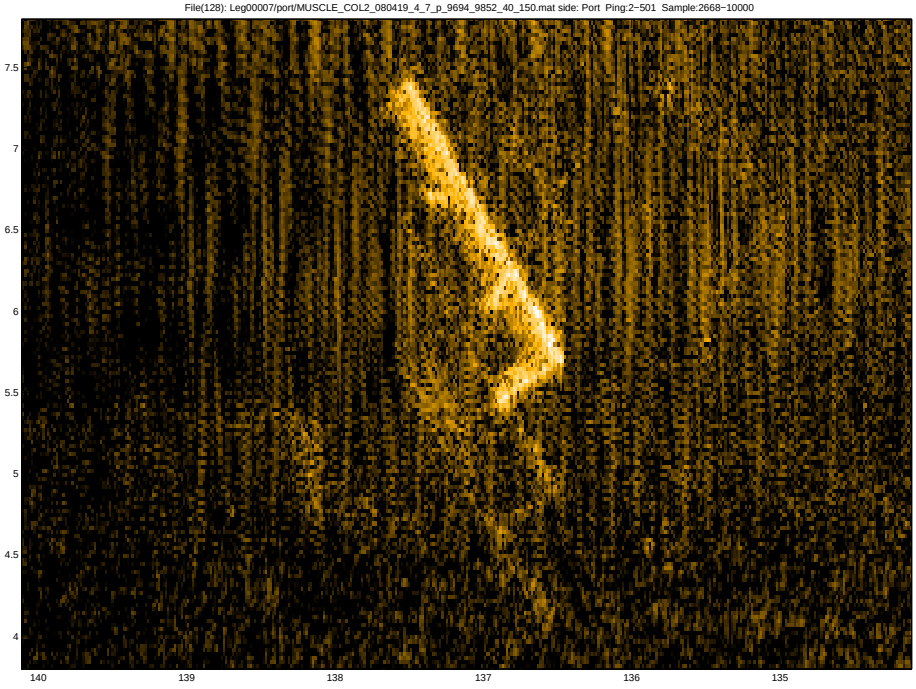


Figure 3.4: Cylinder, problems at imaging platform

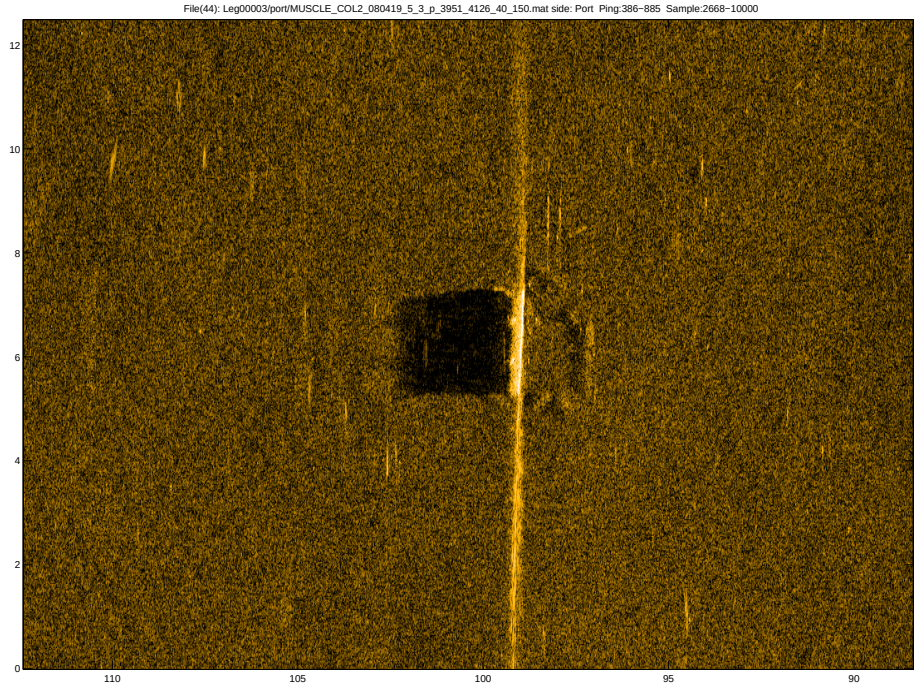


Figure 3.5: Cylinder sideshot

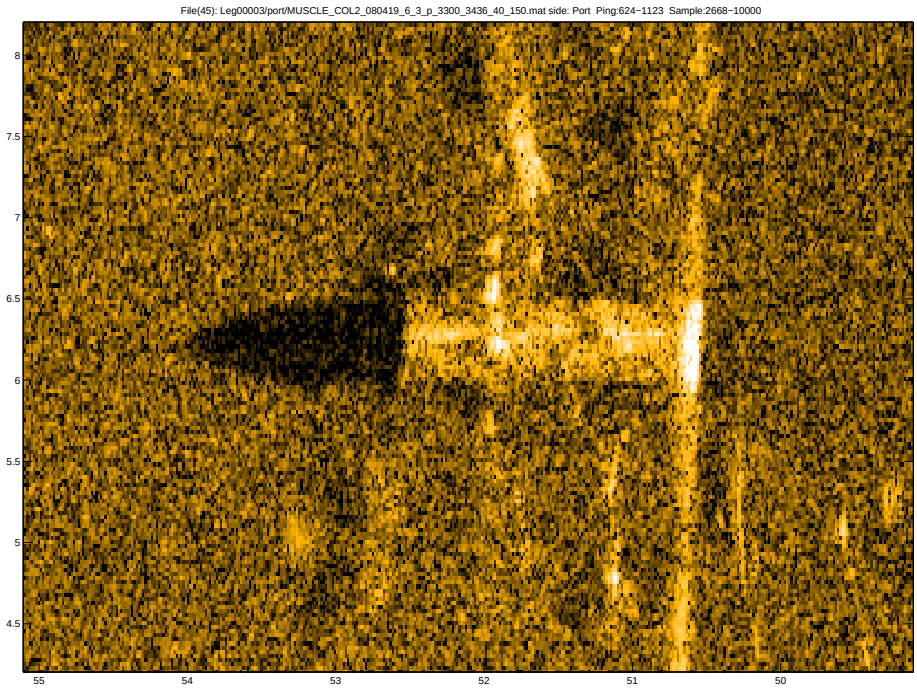


Figure 3.6: Cylinder endshot

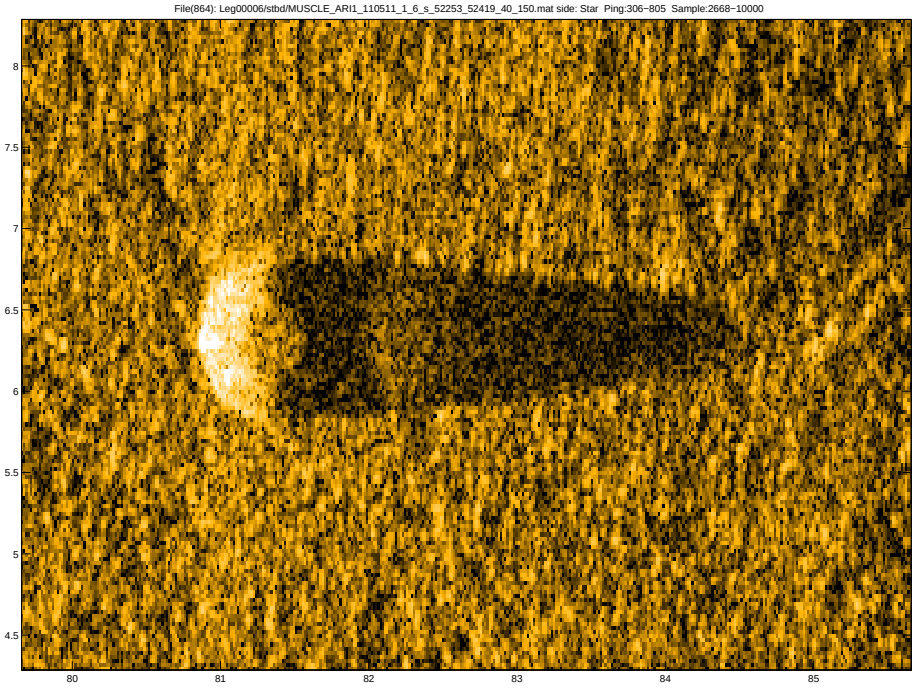


Figure 3.7: Truncated cone

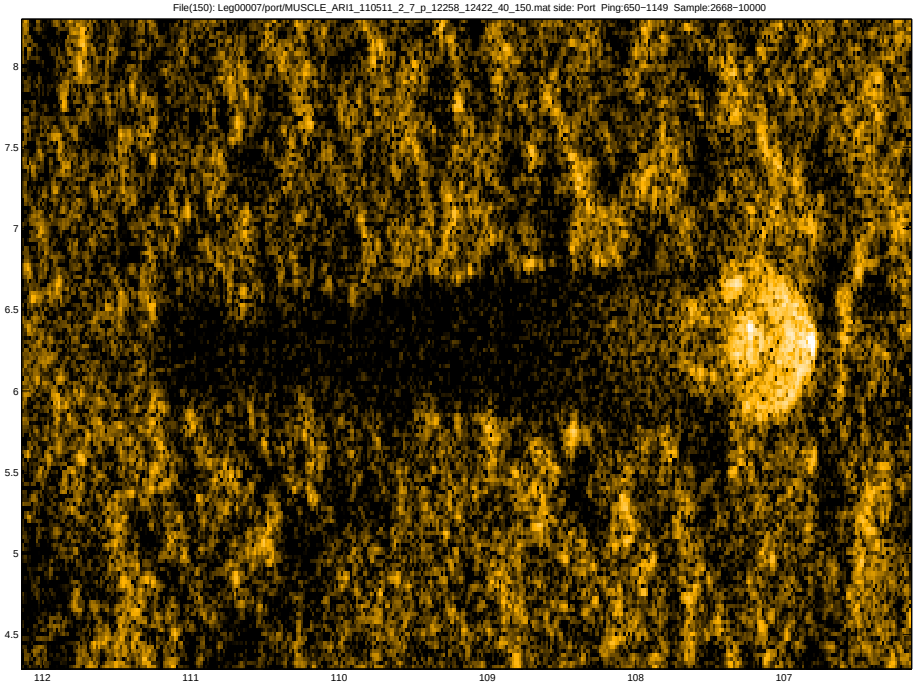


Figure 3.8: Truncated cone

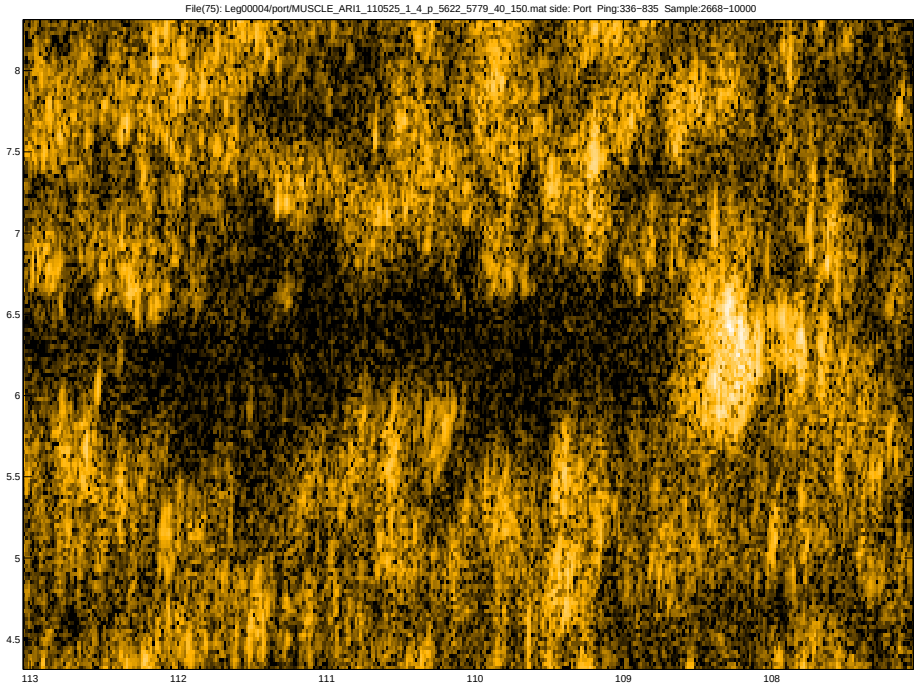


Figure 3.9: Truncated cone on rough surface

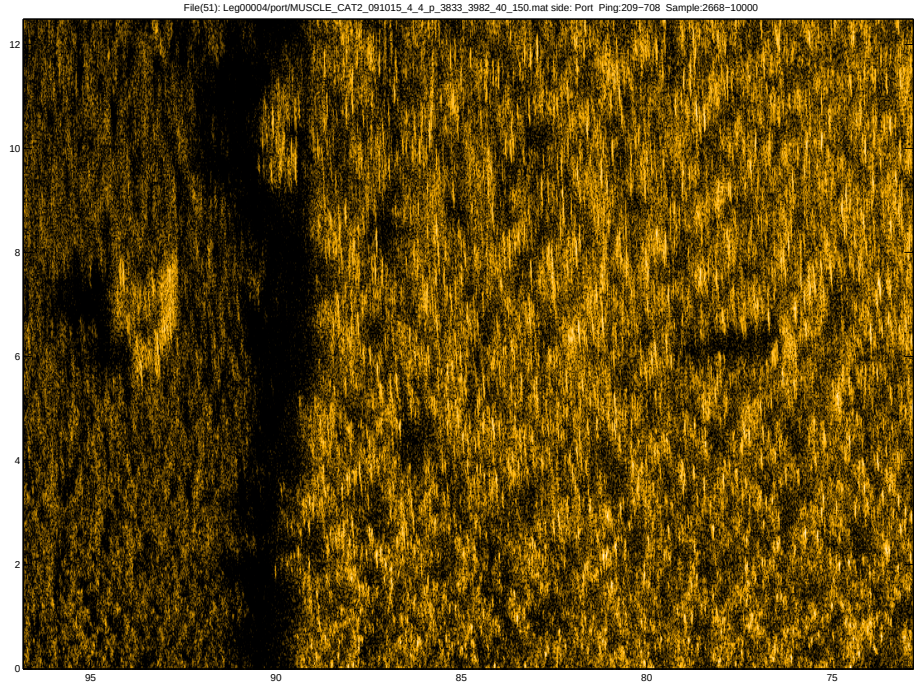


Figure 3.10: Truncated cone among vegetation

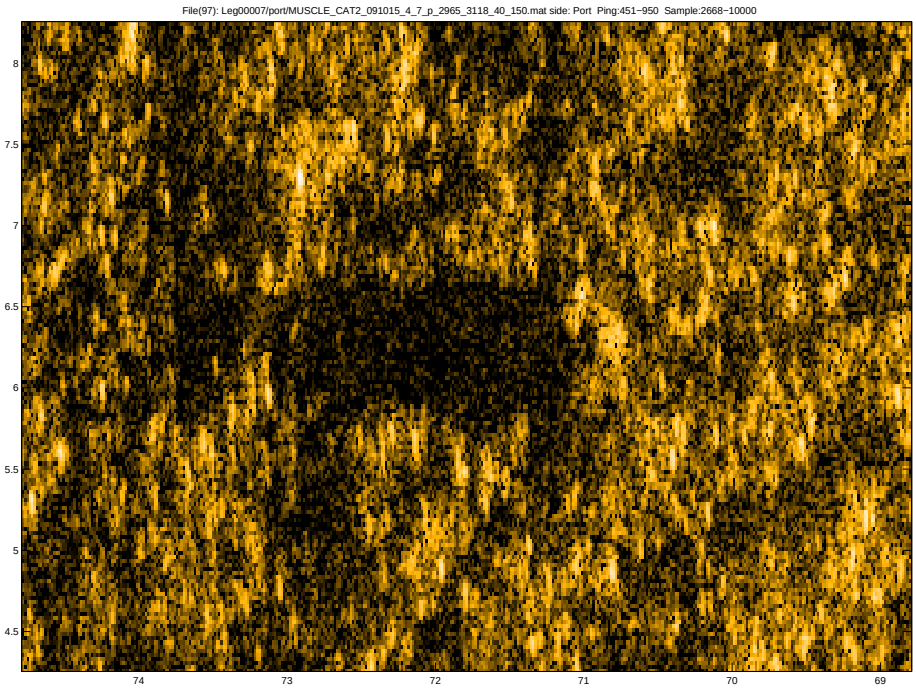


Figure 3.11: Enlarged truncated cone among vegetation

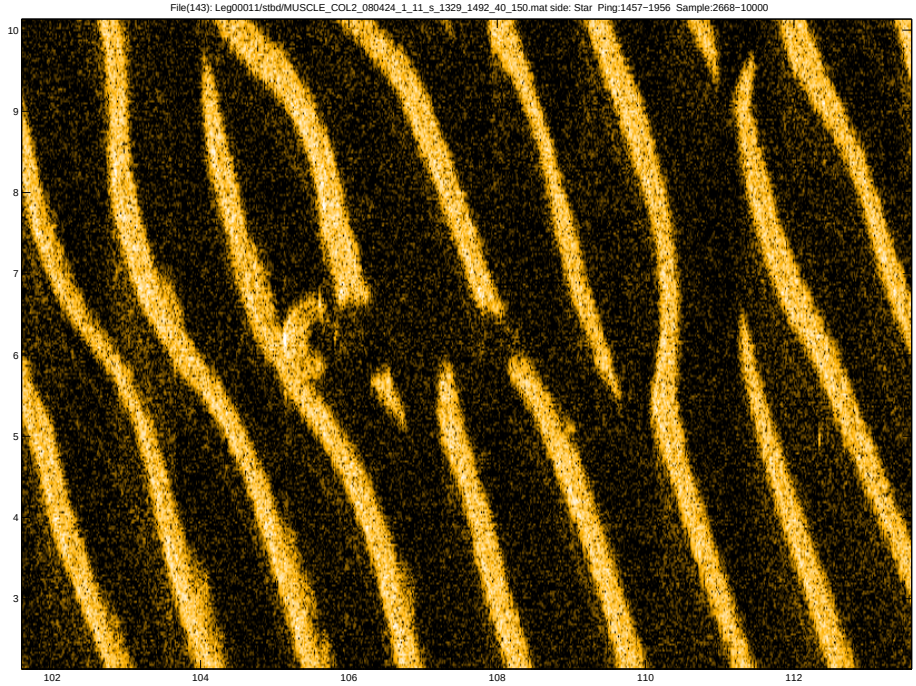


Figure 3.12: Truncated cone on sand ripples

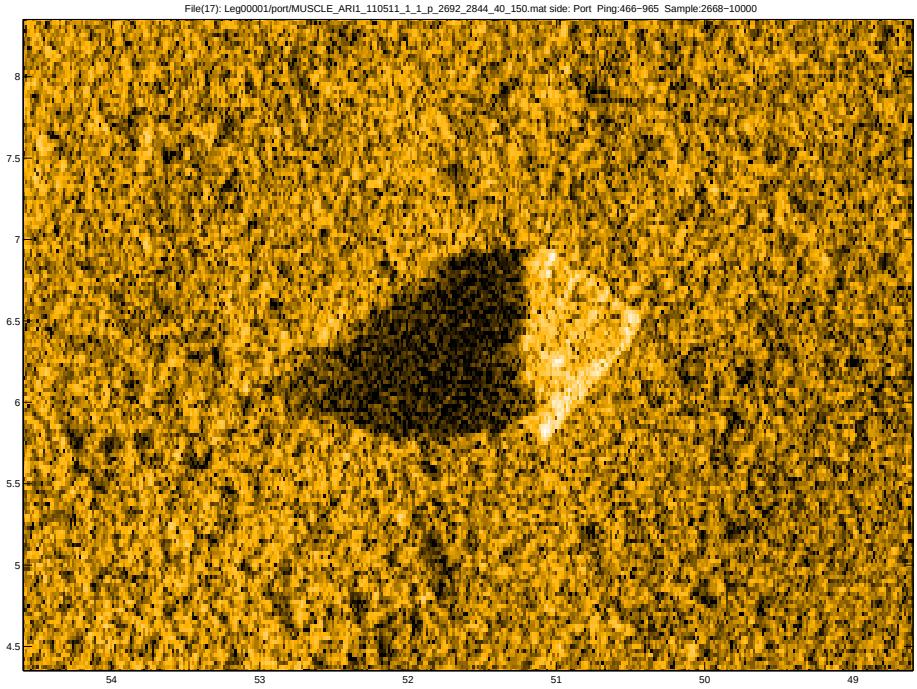


Figure 3.13: Wedge

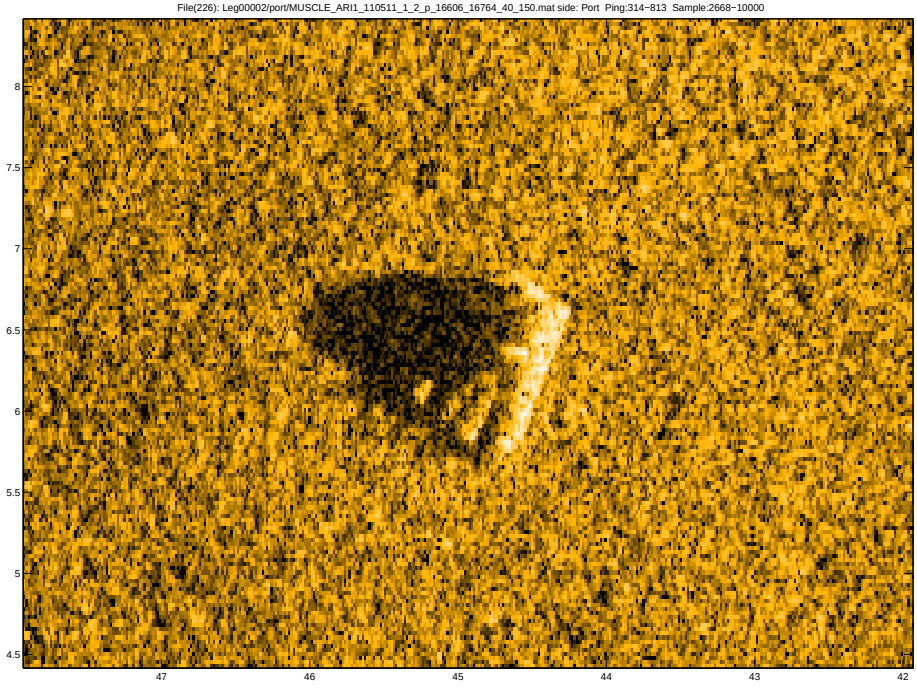


Figure 3.14: Wedge, flipper visible

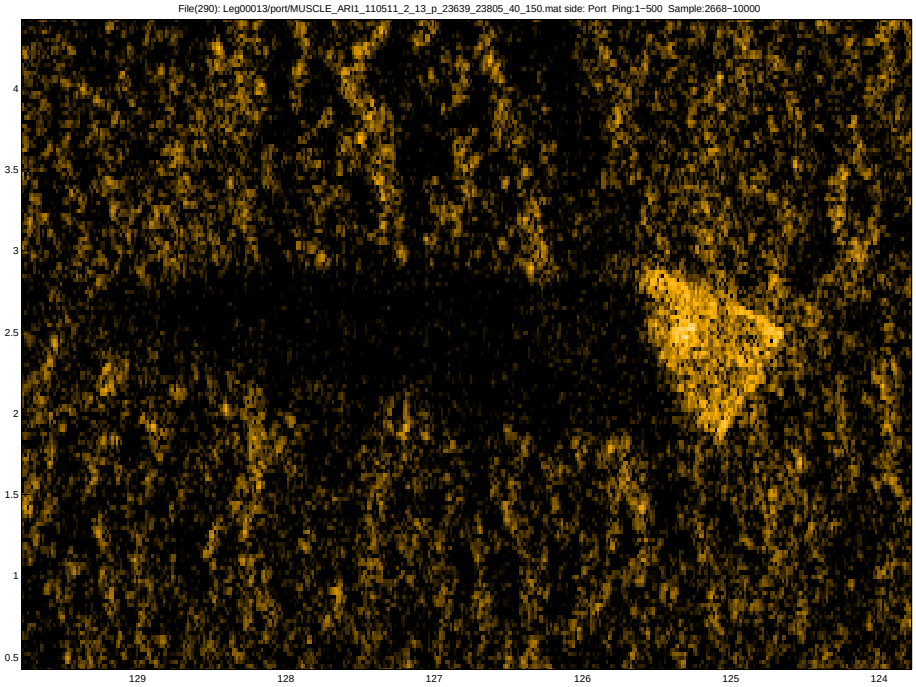


Figure 3.15: Wedge, triangle visible

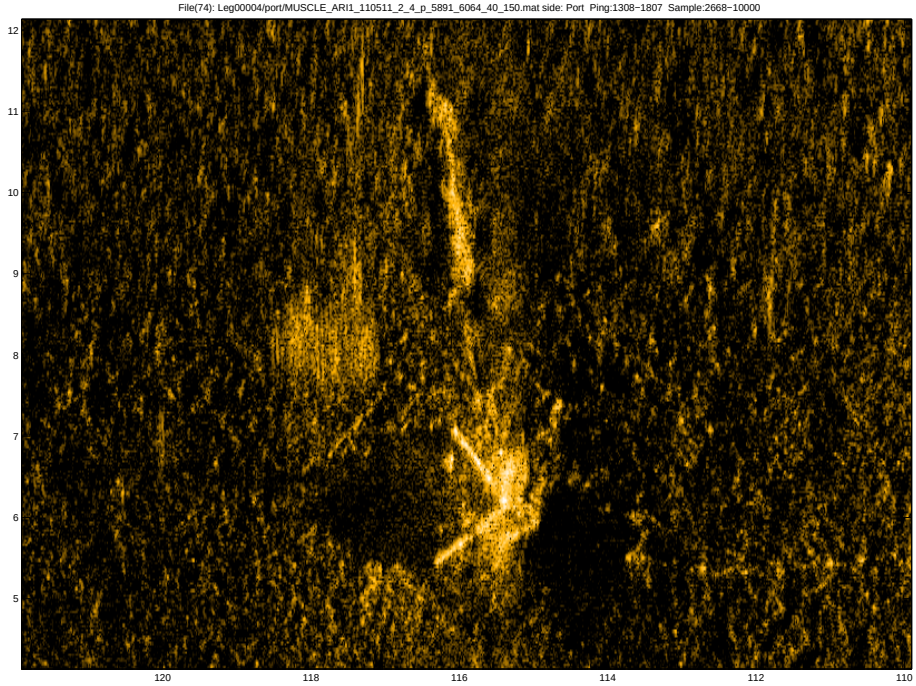


Figure 3.16: Wedge among clutter

Figure 3.17: Large rock

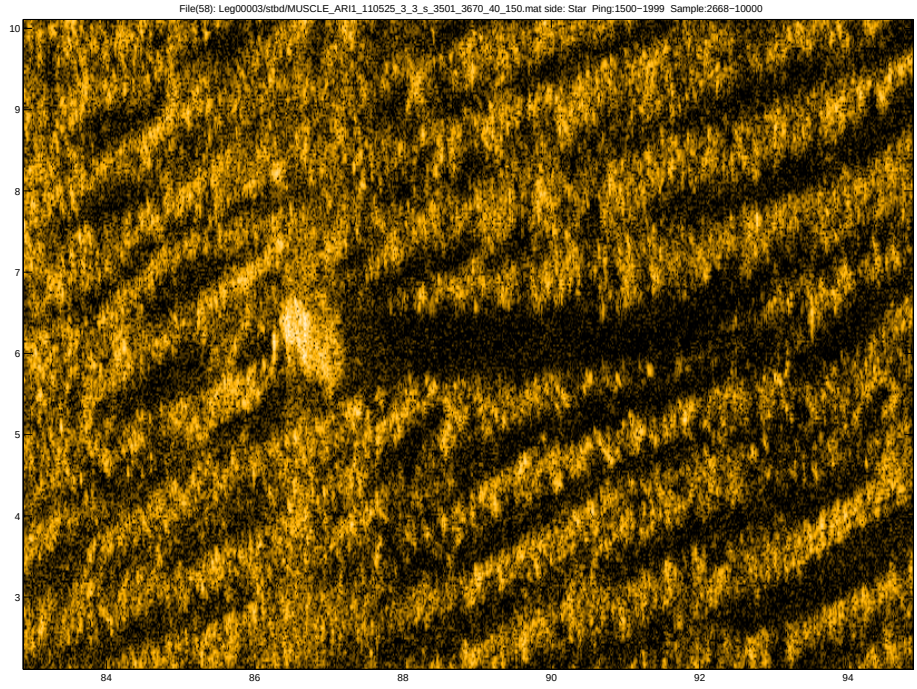


Figure 3.18: Rock on ripples

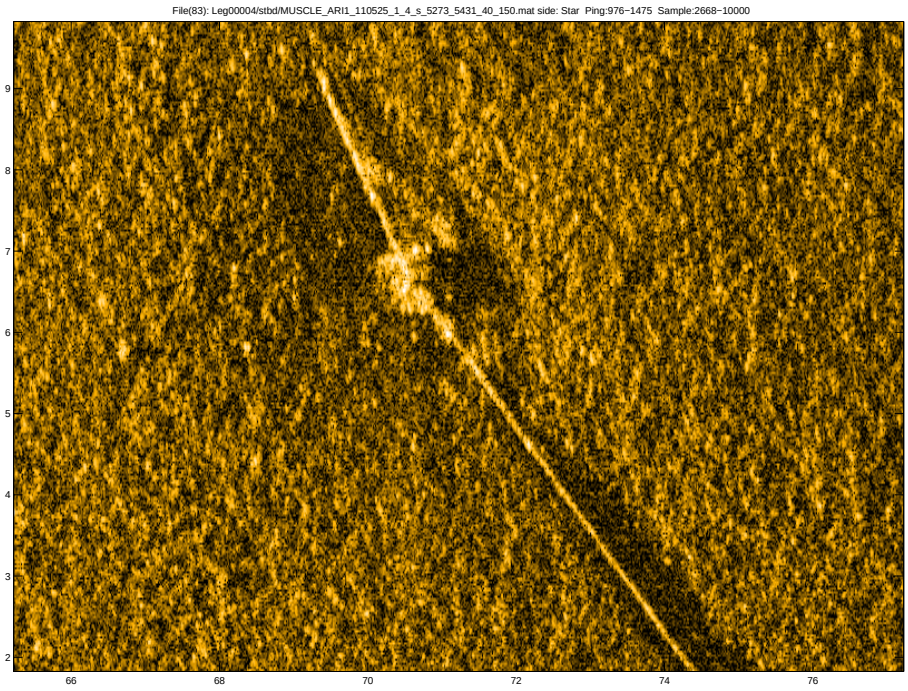


Figure 3.19: Bottom cable

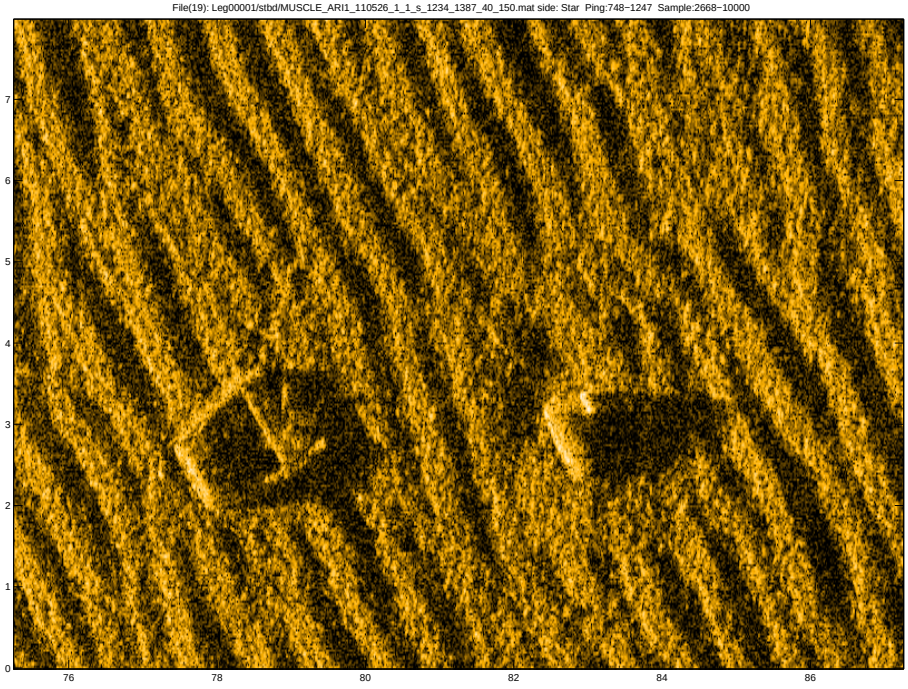


Figure 3.20: Undefined clutter

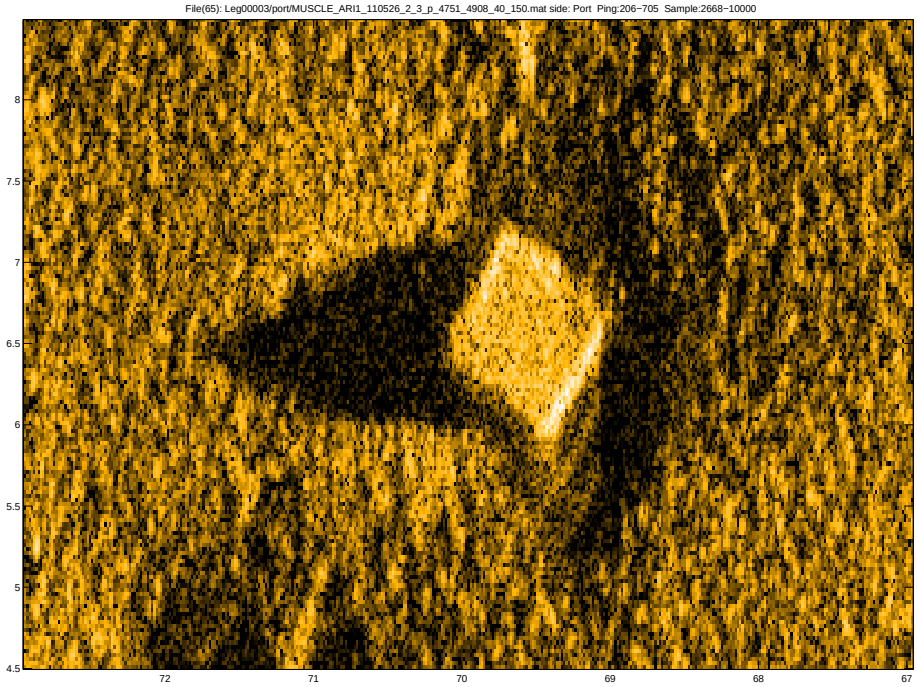


Figure 3.21: Undefined clutter

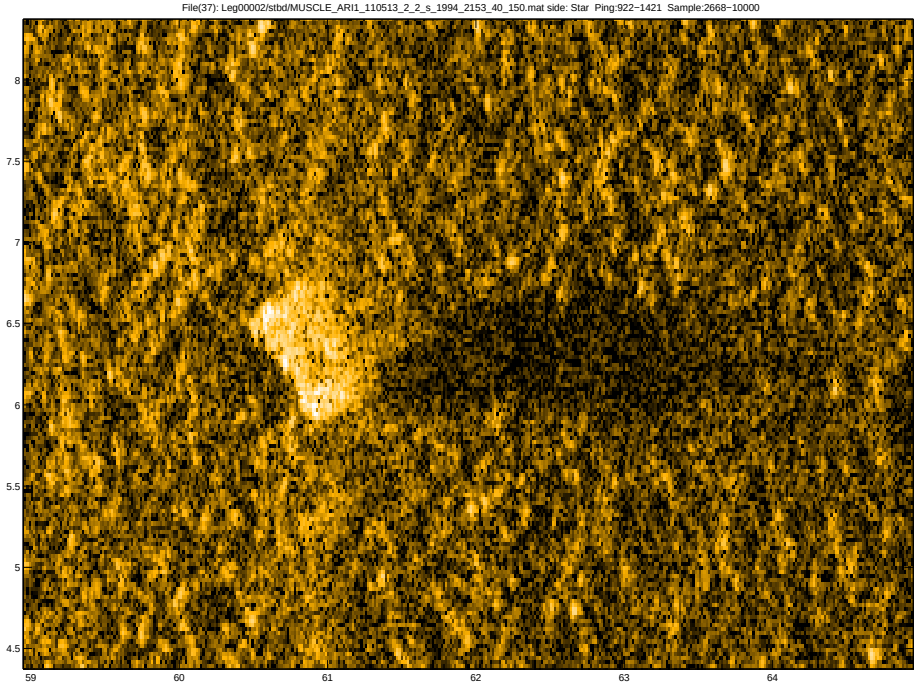


Figure 3.22: Oil drum

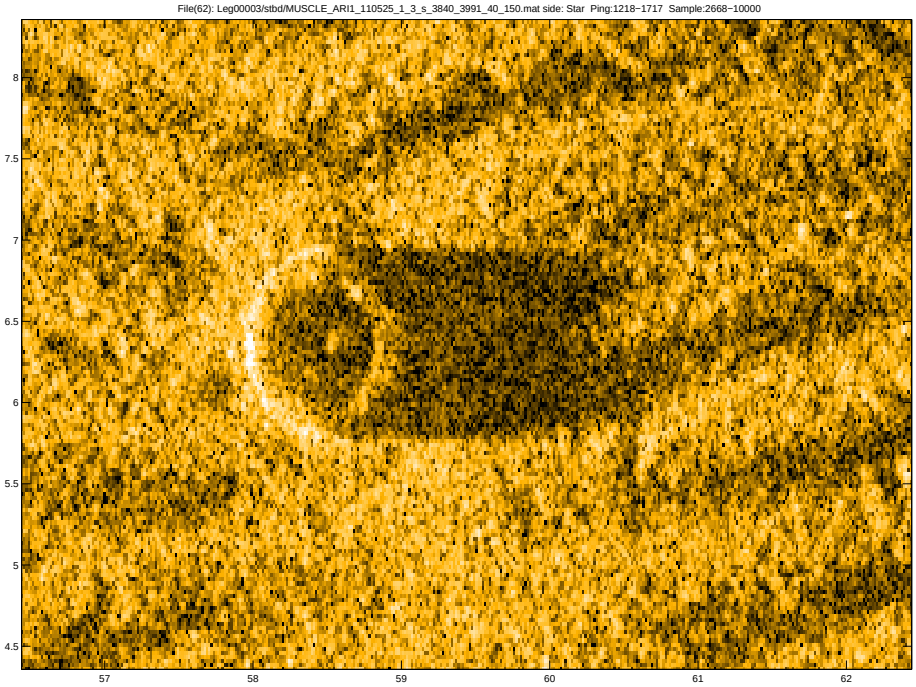


Figure 3.23: Car wheel

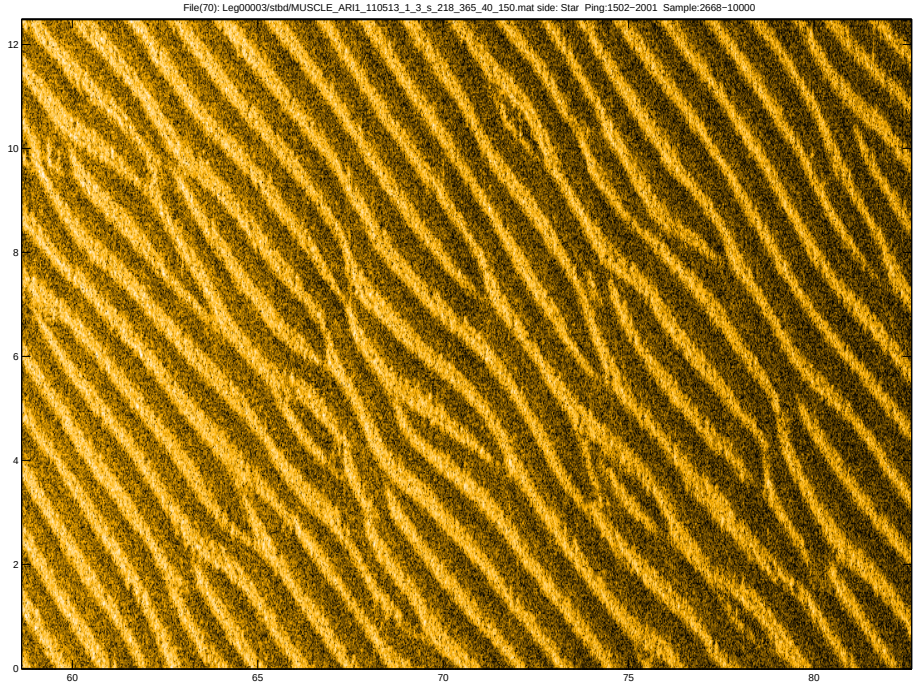


Figure 3.24: Sand ripples

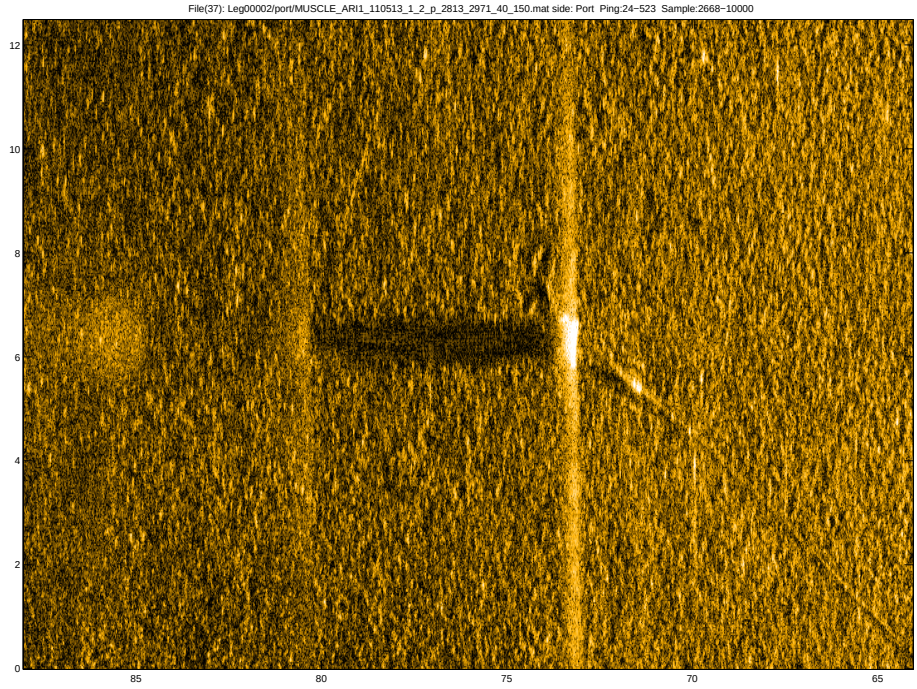


Figure 3.25: Tracks

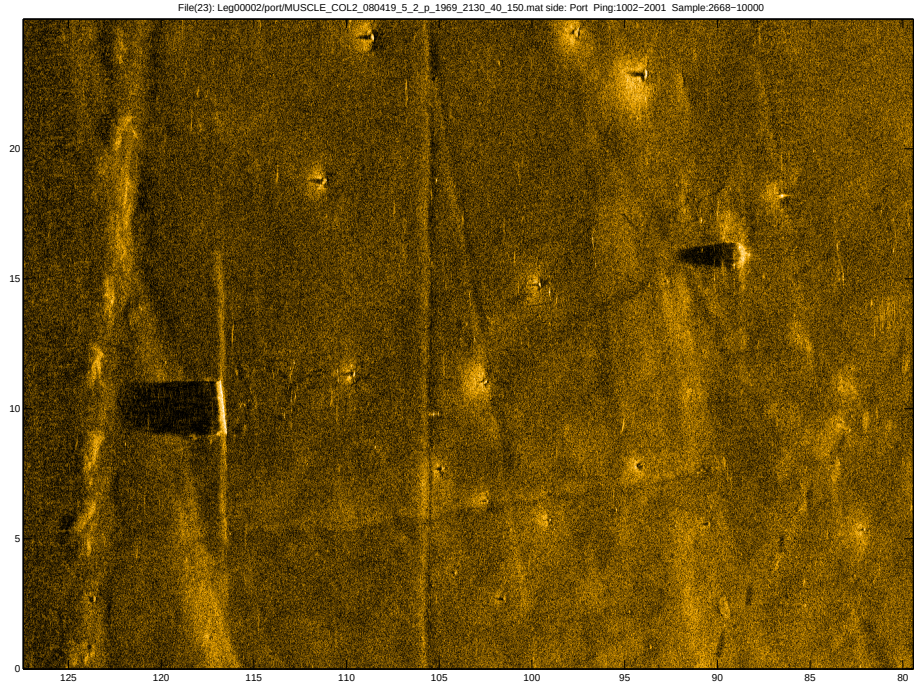


Figure 3.26: Trawl tracks

Chapter 4

Methods

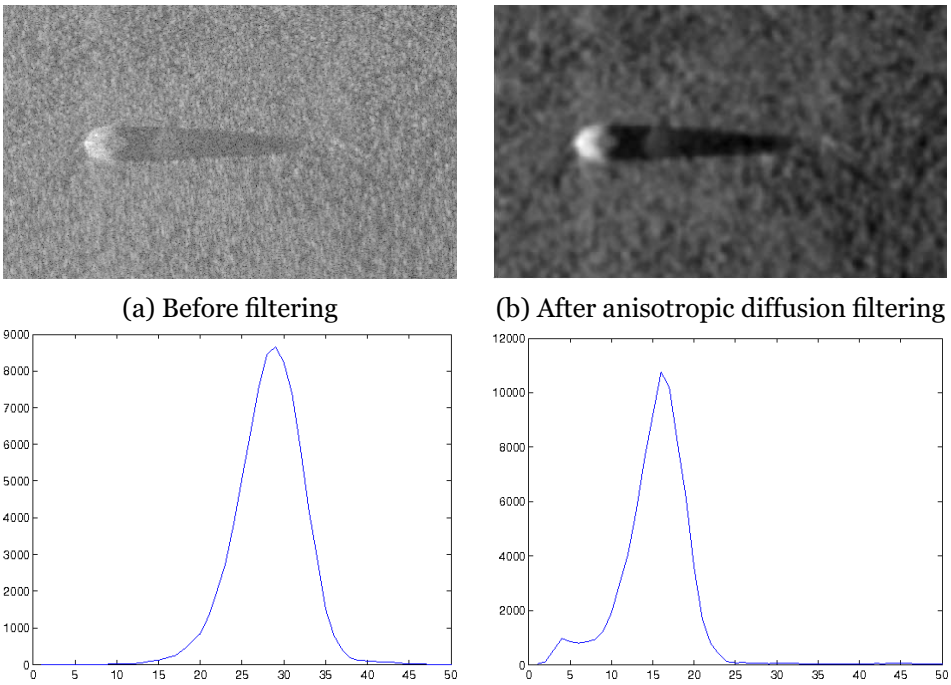
There are two types of methods described in this chapter. The two first ones are filters designed to reduce noise while preserving edges making segmentation easier. They are meant to be used as a pre-processing step before segmentation. The rest of the methods are different kinds of segmentation methods based on statistical and mathematical operations of varying complexity. All considered methods are described in this chapter but only a few of them were tested and have results presented in chapter 5 below.

4.1 Anisotropic diffusion

Noise in images make it difficult for the segmentation algorithms to clearly find edges. Blurring filters will remove noise but tend to blur the edges and diffuse borders between regions in the image as well, and as such will not help. The anisotropic diffusion filter proposed by Perona and Malik [13] is a more intelligent blurring filter that uses several levels of Gaussian blurring and edge detection to define regions in the image and then prevent the blurring across the borders between the regions. In this way edges are not diffused while the detected regions are blurred individually resulting in removal of noise while preserving edges. The filter can be configured with four parameters:

- niter : Number of iterations to repeat the filter.
- K : Local edge contrast.
- λ : Speed of diffusion ($\lambda \in [0, 1/4]$)
- mode : Type of diffusion coefficient function.

Niter specifies how many times the algorithm will repeat the filter, more iterations gives a smoother result but too many iterations will blur too much. K, also called conductance parameter, specifies the local contrast (difference in image intensities on the left and right) of the edges to detect. An edge is defined as a boundary between visually different regions in the



(a) Before filtering

(b) After anisotropic diffusion filtering

(c) Histogram of image before filtering

(d) Histogram of image after filtering

Figure 4.1: Sonar image from Figure 3.7

image. λ defines the speed of the diffusion for each iteration in the 4-nearest-neighbors discretization of the Laplacian operator. It needs to be between 0 and 1/4 for the numerical scheme to be stable. Mode selects one of two diffusion coefficient functions where the first privileges high-contrast edges over low-contrast ones, the second privileges wide regions over narrower regions.

All tests below are performed on images smoothed with this filter. The implementation used is a part of FFI's sonar toolkit called noisefilt. The argument values used for the filter in all the following tests are: niter = 5, K = 25, λ = 0.25 and mode = 2. The values were chosen after a series of trial and error with many different values, the result is an image where noise is greatly reduced while the important features are still present. See Figure 4.1 to see the effect of the filter. Also notice how the filter affects the histogram of the image. The histogram of the filtered image now has two peaks where the largest one marks the mean value of the background pixels and the smaller one to the left marks the mean value for the shadow pixels. The mathematical thresholding method described in chapter 4.8 below builds upon this observation.

4.2 Auto-correlation

In an article about an active contour model [16] the authors describe a method to reduce noise in the image by using something they call Normalized Accumulated Short Term Auto-correlation (NASTA). This

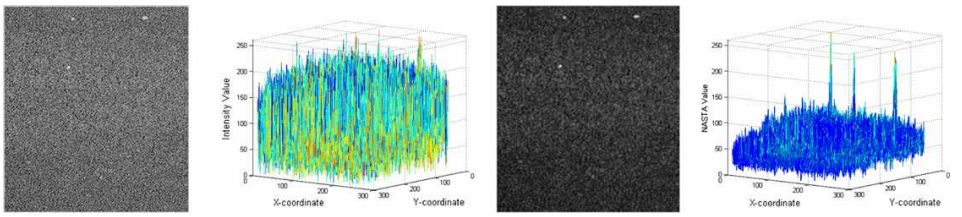


Figure 4.2: From left to right: (a) Original image (b) Surface plot of intensity values of original image (c) NASTA feature image (d) Surface plot of NASTA values [16]

method is based on a signal processing technique originally described by Rabiner and Schafer [14] in a book from 1978 which they called Short Term Auto-correlation (STA). This STA method was designed for speech recognition purposes and the authors of the NASTA method have built upon this technique to make a 2D implementation for images. The NASTA method works by scanning through the entire image and calculate the short time auto-correlation for every pixel with an area around the pixel. Auto-correlation is calculated at a range of different distances and the resulting images for each distance are summed together and normalized to produce a normalized, accumulated STA. The result is a new image of equal size as the original image where areas that are statistically different to their surroundings are enhanced and high frequency noise is suppressed (see Figure 4.2). The original implementation was designed for noisy images with small scattered dots and the goal was to segment these small scattered dots from the noisy background.

The NASTA implementation consists of nested iterations of auto-correlation at different correlation distances in two dimensions. There are many calculations for each pixel and it is computationally intensive. As we can see from Figure 4.3 pixels that are statistically different from the majority of pixels in the image are enhanced while noise is greatly suppressed. One drawback with this method is that it enhances highlight and shadow equally making them all bright pixels in the resulting image. This is because highlight and shadow are both equally different, statistically, from the majority of pixels in the image. A separate step is therefore needed to specify which enhanced region is highlight and which is shadow.

4.3 Representation error

In an article about segmentation of salt diapirs [2] the authors apply a segmentation algorithm based on representation error to smooth a classification result. The salt structures in the article (see Figure 4.4) are round in shape and since the algorithm is designed to find the location of an edge in a line the image is first transformed into a polar representation so that each line crosses an edge.

The algorithm works by scanning trough the polar image line by line

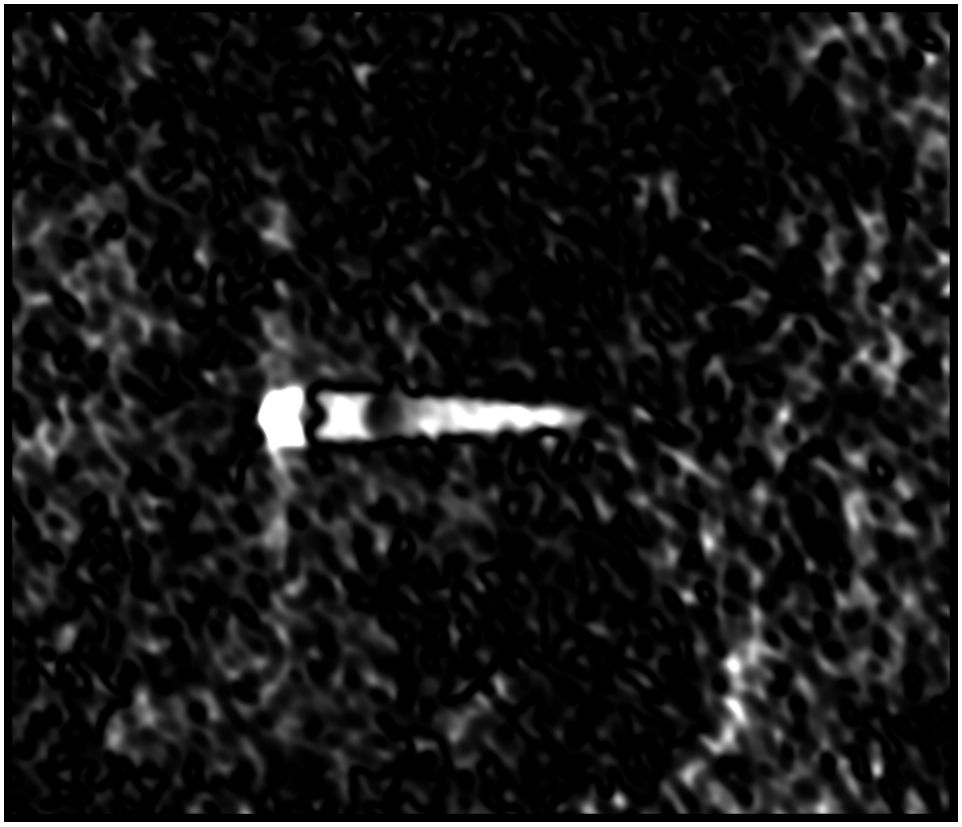


Figure 4.3: NASTA image of the object from Figure 3.7

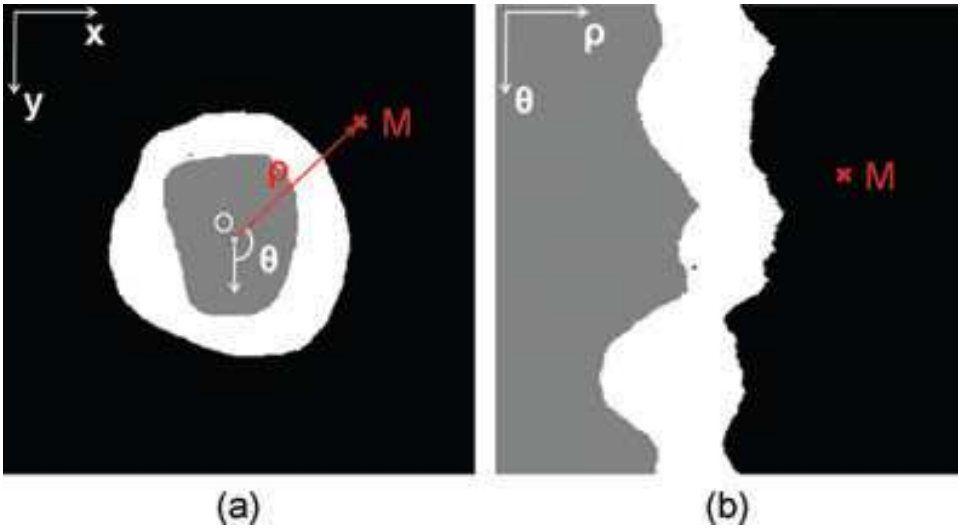


Figure 4.4: Schematic of a time-slice centered around the salt represented in the Cartesian domain (a) and the polar domain after transformation (b) [2]

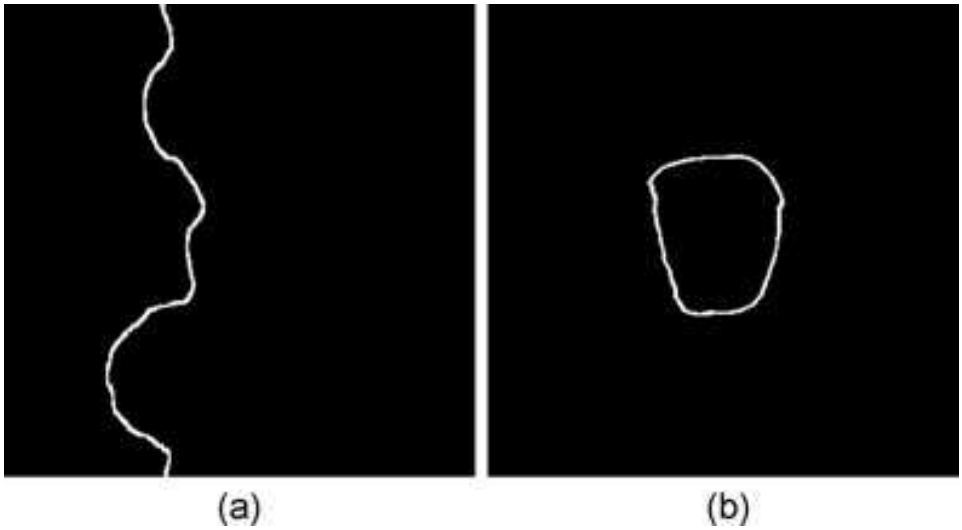


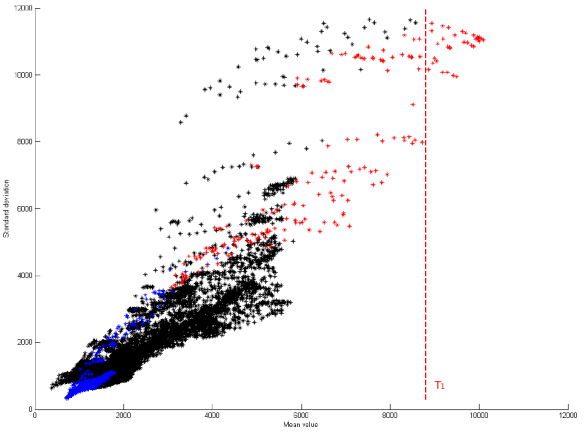
Figure 4.5: Border found by graph-cut segmentation in the polar domain (a) and transformed back to the Cartesian domain (b) [2]

and for each pixel in a line it uses an energy function to calculate the representation error between the right and the left side of the pixel and put this value into a new image on the same pixel location. This new image will have low values where there is an edge and high values where the image is continuous. The most likely location of the edge is found at the lowest values in this new representation-error image.

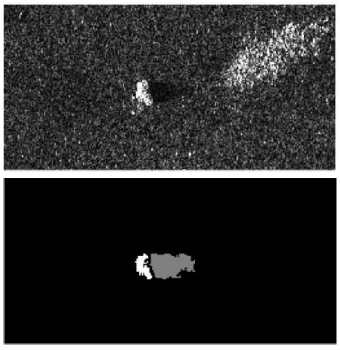
The algorithm assume that the border between two regions will follow a smooth path with no sharp corners. A graph-cut algorithm [18] is then applied to find the smoothest path through the representation-error image. It takes into consideration the location of the edge from the previous line in a cost function and adds penalty for deviation from this path. The result (see Figure 4.5a) is a smooth path from top to bottom following the edge between two regions in the image. This path is then transformed back into the Cartesian domain to get the final segmented region as seen in Figure 4.5b.

4.4 Fuzzy logic

In the article by J. Fawcett *et al.* [6] the authors use a fuzzy logic approach described by Myers [11] to separate echo and shadow from the background. It is an iterative technique based on the fact that pixels of the same class tend to group together and it considers the pixel value and the class membership of the surrounding pixels. It uses a 2D fuzzy membership function with a likelihood that increases with the number of neighboring pixels of the same class but decreases as the pixel intensity value moves away from the typical intensity value of the class. The process repeats until the number of new included pixels drops below a certain value. The largest highlight and shadow regions are usually taken to be the segmented object,



(a) The mean - standard deviation plane for test image 3. Red points are highlight, blue for shadow and black for background. The red line marks the threshold for highlight seed pixels



(b) Test image 3. Rock at range 53m, altitude 10.9m. Top: Original image Bottom: Segmented image

Figure 4.6: Images from the article [4]

but the distance to the detection center is also taken into account. The resulting image is then smoothed with a 5x5 spatial filter and then a two-dimensional 3x3 median filter. This is not a very complex method but slow because of it's many iterations. Also if the image is hard to segment it can end up in an oscillating loop where many pixels change class membership in one iteration and then change back in the next iteration and the algorithm needs to be implemented to avoid unnecessary iterations or in worst case an infinite loop.

4.5 Region growing

The region growing algorithm, as described by J Engstöm et. al [4], works by first identifying a few highlight pixels and then grow this area to include the entire highlight. Then the same is done for the shadow with the addition of limiting it's growth to not exceed the width of the highlight. The initial highlight pixels, also called seed pixels, are found by defining a threshold T_1 (see Figure 4.6a) estimated by a linear combination of the mean value for all pixels in the original image and the maximum value in the local mean image. Then this area is expanded by incorporating neighboring pixels with a pixel value above a lower threshold T_2 that is estimated by a linear combination of the mean-value of pixels already classified as highlight and the mean-value of all pixels in the image. This process is repeated until no more pixels can be incorporated. Holes, or non-classified pixels surrounded by pixels classified as highlight, are then filled.

The seed pixels for the shadow region is found by defining a window behind the classified highlight and calculating mean and standard deviation of the pixels within the window. Initial shadow pixels are selected by using a mean value threshold T_3 and a standard deviation threshold T_4 on the

pixels within the window. The resulting region is then grown using the same principles as for the highlight, the only difference is that the shadow is not allowed to grow wider than the width of the highlight. The final result (see Figure 4.6b) is two regions describing highlight and the shadow.

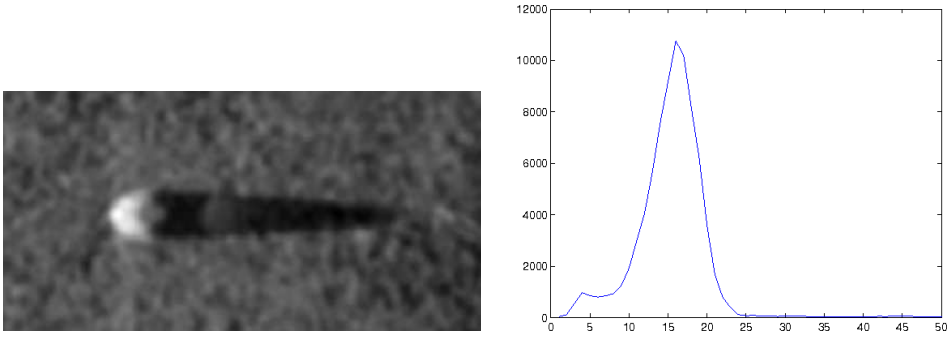
4.6 Markov random fields

Markov Random Fields (MRF) is a statistical segmentation method that uses *a priori* information about the intensity values of different classes, such as shadow and background, and the intensity values of neighboring pixels to label all pixels as belonging to one of the predefined classes. MRF can be used in different ways; Ferdandos and Zoubir [5] use MRF on a downscaled version of the image to roughly find the location of highlight and shadow and then initiate a snake algorithm. Mignotte *et al.* [10] describe in detail a way of segmenting object shadow from sea-bottom and comparing several different methods of parameter estimation. Reed *et al.* [15] uses MRF as a detection stage where possible objects are detected in large sonar images and later segmented more accurately by a snake algorithm.

The main difficulty of unsupervised Markovian segmentation is the estimation of parameters required for the segmentation. These parameters vary from case to case and depend on factors in the image such as noise, bottom type, sonar resolution etc. and to make a robust MRF segmentation algorithm these parameters need to be estimated automatically. The Iterative Conditional Estimation (ICE) algorithm is a method used by the authors mentioned above as a way to estimate these parameters. The *a priori* knowledge needed to label the pixels are pixel density functions (PDF) defining the mean and standard deviation of the pixel intensity values for each class. The PDFs are used in combination with the values of the neighboring pixels and the parameters from the ICE to calculate a Markovian probability. This is the probability for the pixel to belong to any of the classes. The pixel is then labeled with the most probable class. This process is repeated iteratively. ICE is based on data from the labeled pixels and the Markovian probability is calculated based on the ICE parameters and so the algorithm alternates between these calculations. The process is repeated until the amount of pixels changing label is below a certain threshold or until exceeding a maximum of iterations.

4.7 Snakes

Statistical active contours, or snakes, is a different way of segmentation that assumes that the regions to segment have a continuous edge. The snake can be imagined to be like a rubber band consisting of a finite number of points that have tension and stiffness that will define its ability to form sharp corners and cause it to contract or expand. It also contains a function to attract it to certain features in the image that constitutes the edge between



(a) Small region of object (b) Histogram of image

Figure 4.7: Sonar image from Figure 3.7

two regions to segment. This attraction is defined by an energy function that will differ depending on application. The snake moves by recalculating the optimal position of the points in the snake iteratively. If a position with higher energy is found the point is moved and iteration continues until no more points are moved or until a set iteration limit is reached. This iterative nature of the method makes it computationally intensive. Another drawback of the snake is that it needs an initial position or contour to start moving from. If the object to segment is not within the initial region the segmentation process will fail. This is why a snake segmentation often is used after a previous segmentation operation where the objective is to find the location of the object. In the article by Fandos and Zoubir [5] the authors use Markov random fields (MRF) to find the initial region to initiate the snake. They configure the MRF to be fast but not very detailed. This makes the calculation quick and gives a rough region as a good starting point for the snake.

4.8 Mathematical thresholding

This method is based on simple mathematical operations to form a fast and simple way of finding threshold values for the highlight and shadow and then do some post processing to remove non-relevant segmented regions. By looking at the histogram of the filtered image (see Figure 4.7b) we see that most of the pixels in the image are in the middle of the histogram. This major group of pixels are the background pixels. The shadow pixels in this image are below 7 and the highlight pixels are above 25. If by some simple calculations we could extract the start and end of the largest peak in the histogram it should be possible to segment the three classes. The algorithm proposed here and explained in more detail in chapter 5.3 below uses the histogram and its first and second derivative to find values for thresholding. After thresholding any additional small blobs of pixels are removed and if there are several shadow blobs the ones not behind the highlight are removed.

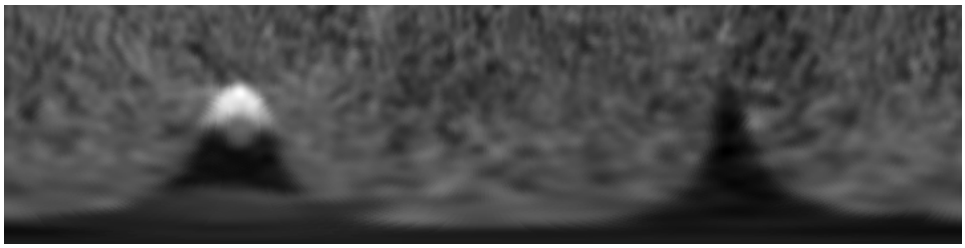
Chapter 5

Results

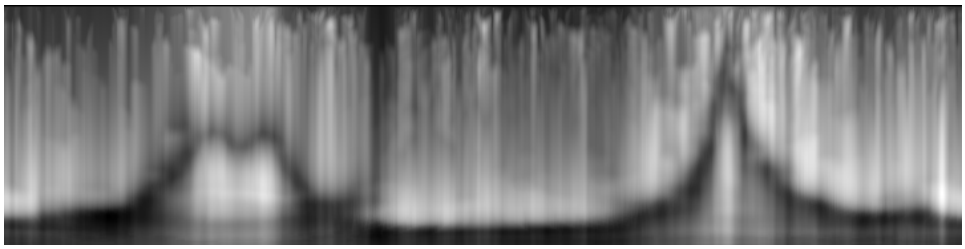
In this chapter the tested methods are presented with their individual results. Only some of the methods mentioned above were chosen for further testing.

5.1 Representation error

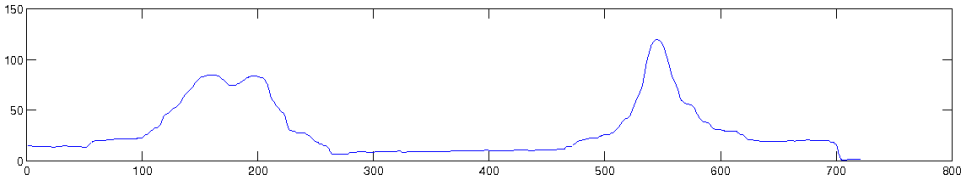
The representation error algorithm works by analyzing the pixel values on both sides of an edge and then finding the most probable location of the edge. It was tested on the shadow only and not the highlight because the shadow is more difficult to distinguish from the background. Since this method expects an image where all lines cross the edge a polar transform is needed. To do a polar transform a center point must first be selected to rotate around. This point were in these tests manually chosen in the middle of the shadow, or if the shadow was cone shaped it was placed in the middle of the largest part of the shadow. The results of the polar transform can be seen in Figure 5.1a. Calculation of the representation error between the two sides of the edge (top and bottom in this case) were then calculated and the result can be seen in Figure 5.1b. The graph-cut algorithm then calculates the best path taking into consideration the location of the edge from the previous line. The path is shown in Figure 5.1c. This path is then transformed back to the Cartesian system and the final result is shown in figure 5.1d. It turns out that this algorithm is sensitive to a lot of factors. The center placement is very important, just being somewhere inside the shadow is not good enough because the calculation of the representation-error needs a certain amount of pixels on each side to be reliable. If the center point ends up too close to the edge the algorithm will not be able to find the correct path. If the center point is outside the shadow you will get a result like in Figure 5.10b. The starting angle of the polar transform is also important. A wrong starting angle might cause the graph-cut algorithm to start out in the wrong location causing the whole path to be wrong or in best case it will eventually make its way into the right location but still not produce a good result. If the algorithm fails to find the correct path you will get a result as in Figure 5.7b. Both these problems also vary depending on the image to segment and it is difficult to define the best starting position



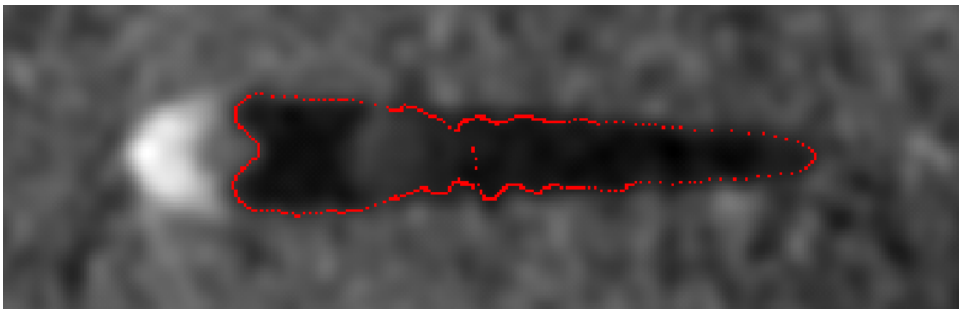
(a) Polar transform



(b) Representation error



(c) Graph-cut path



(d) Final segmentation

Figure 5.1: Sonar image from Figure 3.7

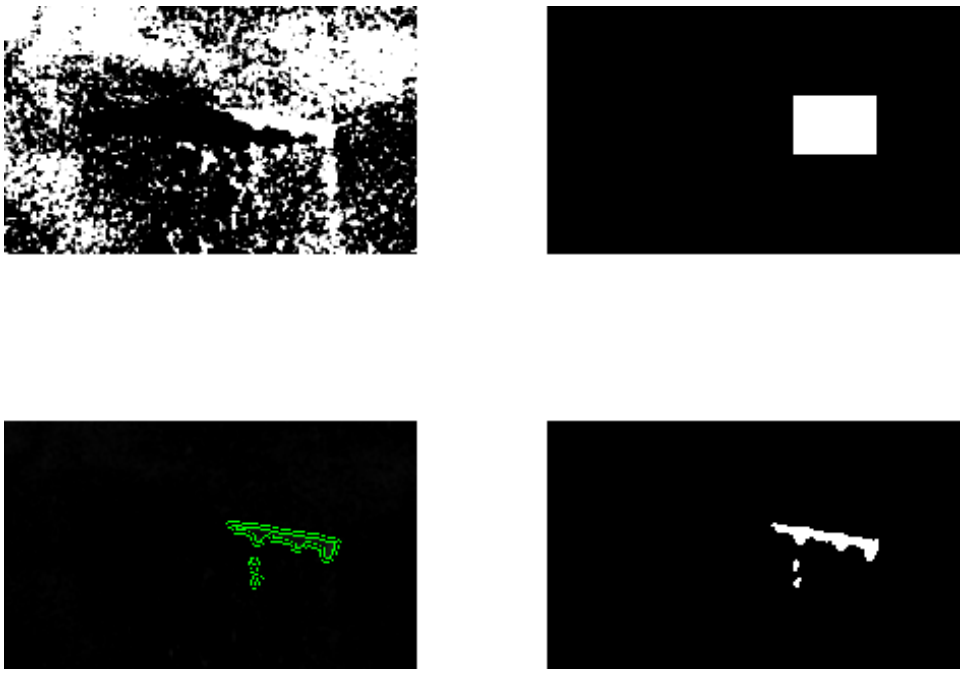


Figure 5.2: Snakes segmentation of the object from Figure 3.3 (highlight)

and start angle. This method is therefore most suitable to follow a previous calculation step maybe to improve the result or smooth the edge of an already segmented region that can give the location of the center point and also a starting point for the graph-cut algorithm.

5.2 Snakes

There are many implementations of the snakes algorithm freely available on the Internet and the decision was made to look for existing implementations instead of implementing one from scratch. After testing several of them one was chosen because of its ease of use and its ability to work on sonar images with only minor adjustments. This implementation is made by Shawn Lankton [9] and is a region based active contour segmentation implemented in matlab. Tests were done to segment the highlight and the shadow (see Figure 5.2 and 5.3) where the initial position was manually placed over the highlight or shadow and number of iterations were limited to 400. As we see from the images the snake is initiated with a square block that covers most of the shape to segment. This means the size and location of the shape needs to be known before the algorithm can be started.

Most snake implementations found on the Internet are designed to work on relatively simple images with clear but complex shapes with some noise and mostly performed on images that have already passed through a thresholding process. The design of the energy function in such implementations are not made to cope with sonar images. There have been some research [8] into building an energy function that operates directly

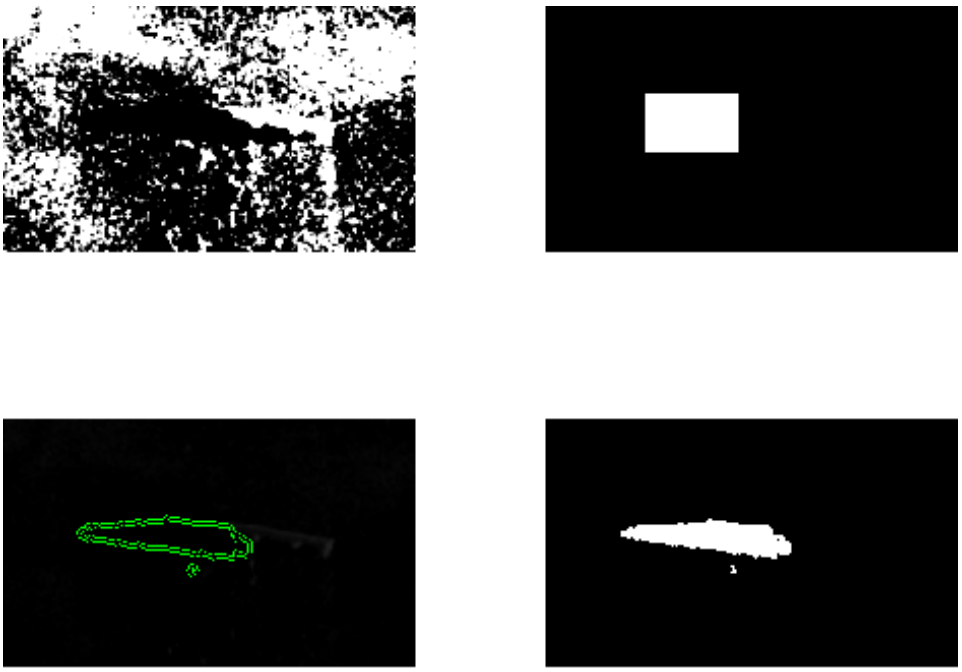


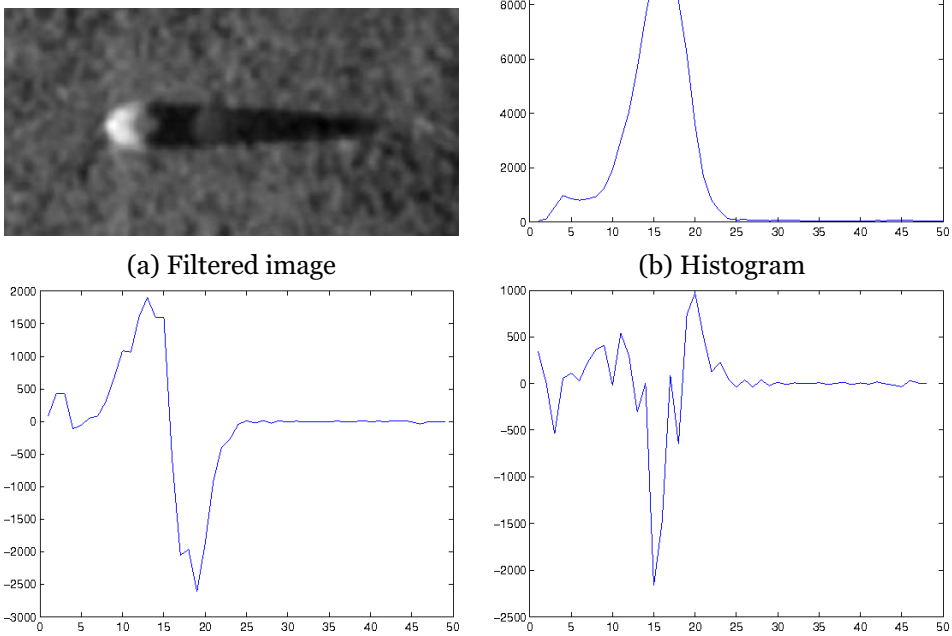
Figure 5.3: Snakes segmentation of of the object from Figure 3.3 (shadow)

on the image without the need for initial processing. This approach would simplify the process by removing one step and making it a pure snakes method. This would make it less computationally intensive however, it would not remove the need for initiation. A way to solve this problem is discussed by Alireza Vard *et al.* [16] where instead of one snake, multiple snakes are evenly distributed all over the image. All the snakes with no features inside will shrink and disappear while the ones covering features will remain. If several snakes cover the same feature they will merge as they grow in contact with each other. This approach will remove the need for initiation but at the cost of adding a lot of calculations. On sonar images where only one highlight and one shadow is the goal for segmentation this multi-snake method is very costly, it would end up segmenting all highlights and shadows in the image and would need a post-processing cleanup stage where unwanted regions were removed.

5.3 Mathematical thresholding

The threshold values are found by doing simple calculations on the histogram and its first and second derivative. A plot of all these is shown in Figure 5.4. The value for the highlight is found by first extracting the location of the highest peak from the second derivative, which in this example is 21, and then adding 20% making the threshold value for the highlight 26.

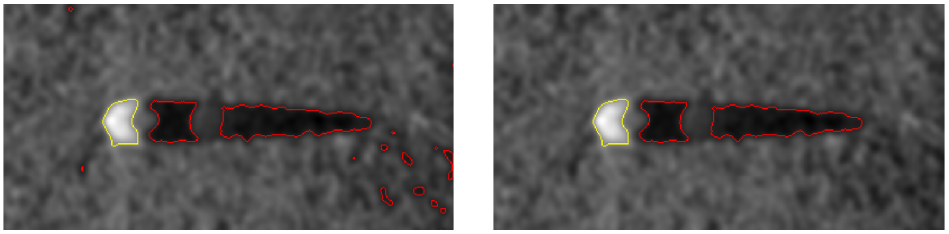
For the shadow the algorithm first looks for a smaller peak to the left of the major peak in the histogram and if found placing the threshold value at



(a) Filtered image (b) Histogram

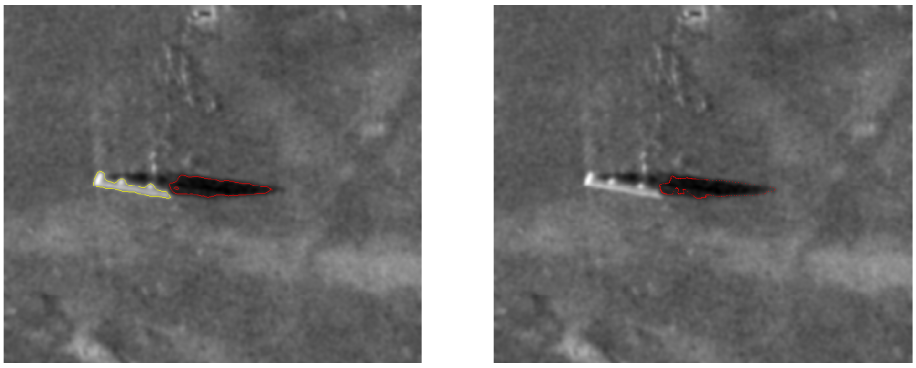
(c) First derivative (d) Second derivative

Figure 5.4: Sonar image from Figure 3.7



(a) Before cleanup (b) Final segmentation

Figure 5.5: Mathematical thresholding of object from Figure 3.7

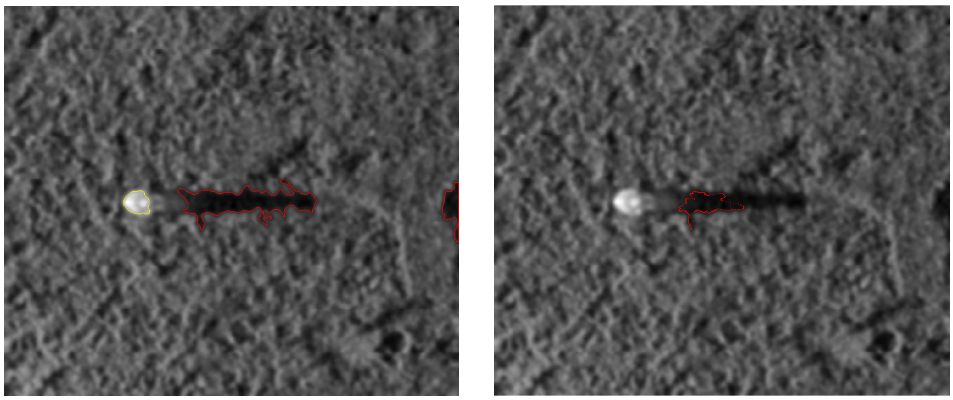


(a) Mathematical thresholding (b) Representation error

Figure 5.6: Segmentation of object from Figure 3.3

40% between this peak and the largest peak in the first derivative. In this case a peak is found at 4 and the peak of the first derivative is 13 making the threshold value 7.6. If there is no shadow peak the threshold value is set by finding the first largest peak in the second derivative and placing the threshold value in the middle between this value and the peak of the first derivative.

The result of the segmentation can be seen in Figure 5.5a. To remove the unwanted segmented pixels of the shadow, all blobs with less than 500 pixels are removed and also blobs of shadow pixels not behind the highlight are removed. The final result can be seen in Figure 5.5b. Even though this method seems to work very well there are several drawbacks. Clearly visible objects on simple backgrounds are not that difficult to segment and therefore gives good results, if there are more vegetation and uneven background, segmentation is not so good (see Figure 5.8). The other problem is that it contains several constants and semi guesses at how much to add or remove from the threshold values to end up with a good threshold value. If the start and end of the major peak of the histogram could be more precisely calculated, maybe by a sort of curve fitting, the results could become more robust and more accurate. Also the size of the window containing the object greatly affect the histogram and a window too large or too small can make the calculations fail.



(a) Mathematical thresholding (b) Representation error

Figure 5.7: Segmentation of object from Figure 3.8

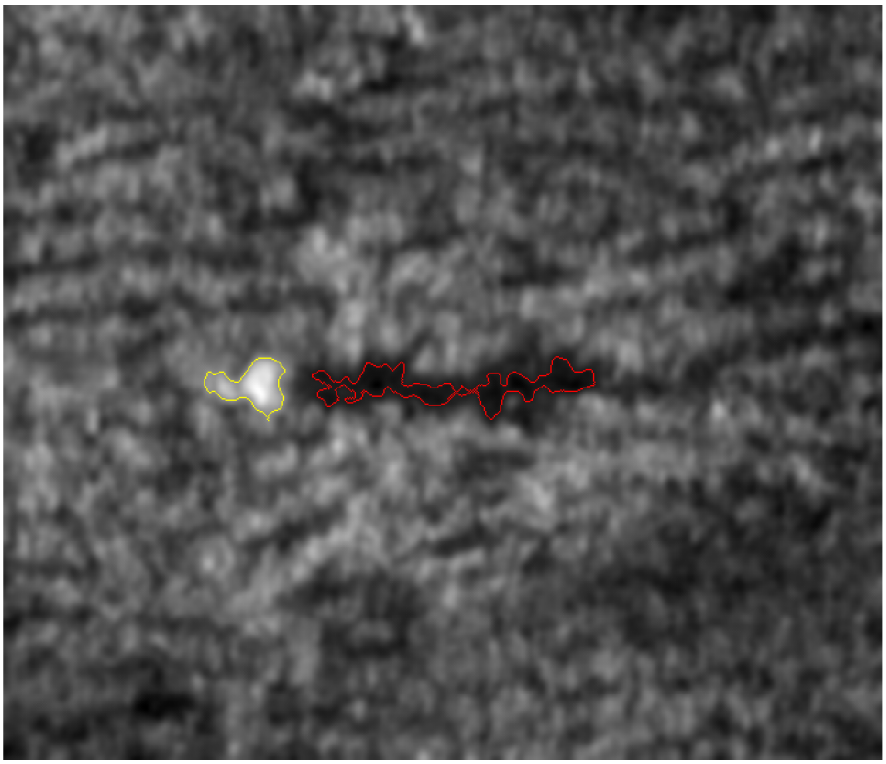
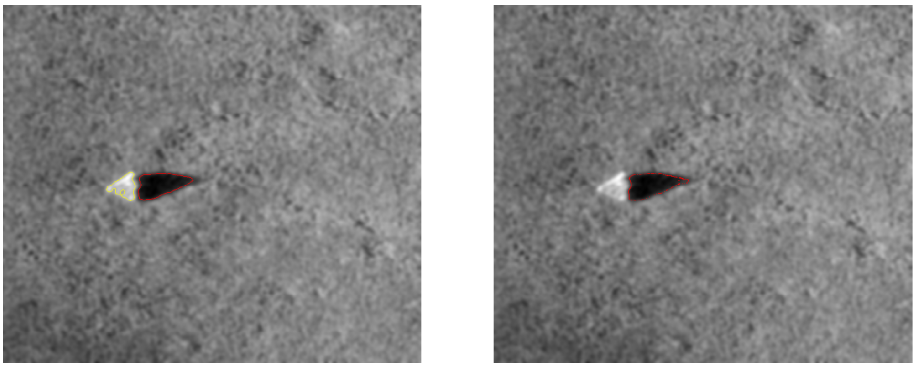
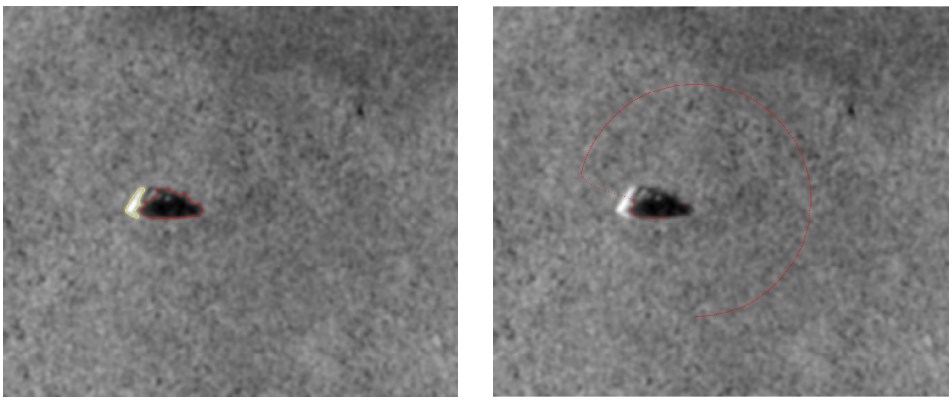


Figure 5.8: Mathematical thresholding of object from Figure 3.9



(a) Mathematical thresholding (b) Representation error

Figure 5.9: Segmentation of object from Figure 3.13



(a) Mathematical thresholding (b) Representation error

Figure 5.10: Segmentation of object from Figure 3.14

Chapter 6

Conclusion

In sonar image segmentation the first problem that needs to be solved is the image noise. Two filtering methods were proposed here were the first one based on auto-correlation did a good job suppressing noise while loosing some of the pixel intensity information. The second method based on anisotropic diffusion managed to filter most of the noise while preserving edges and other image information. The anisotropic diffusion filter was chosen because its ability to preserve important information in the image while removing unwanted noise.

The second issue is the segmentation itself. Here several methods were tested for their ability to identify and separate the highlight and shadow from the background. A method based on representation error proved to be able to find the most probable location of the edge when the edge was difficult to detect. But it is a very unstable method that can easily go wrong if not initialized properly or if the image is not what it expects. This is not good if it is to be used in unsupervised systems where there is no operator to evaluate the result. The method itself has no way of telling if it succeeded or not and it is difficult for any other method to do so as well.

The snakes method suffers from many of the same problems where it needs proper initiation and produces unpredictable results. This unpredictability makes it a less likely choice because we want unsupervised systems to be stable, reliable and predictable with no need for human intervention.

The last tested method based on mathematical thresholding proved that the anisotropic diffusion filter makes segmentation easier and even this less complex method produced very good results in most cases. The complex images however could not be properly segmented and for those cases more advanced methods are needed. The mathematical calculations behind this method still suffers from a lot of constants and could be improved by using more advanced ways of finding the start and end of the largest peak in the histogram producing even better results on a wider range of cases.

Segmentation is hard and even though it looks easy for us to see with our eyes with our brains natural ability to perform edge detection and segmentation our computer algorithms still have a long way to go before matching us. The greatest advantage of us and the brain is that if we

know the shape of the object to look for we can visualize what it will look like image and from the visualized model trace out where we expect the parts of the object to be. Research into computer algorithms working this way [3] is in progress and are starting to show some promising results. We might see more implementations in this direction in the future since these algorithms require even more computing power because it uses 3D projection techniques to visualize virtual objects for comparison with the real ones in an attempt to get a match.

Bibliography

[1] Andrea Bellettini and Marc Pinto. Design and experimental results of a 300-khz synthetic aperture sonar optimized for shallow-water operations. *IEEE Journal of oceanic engineering*, 34(3):285–293, 2009.

[2] Angélique Berthelot, Anne H. S. Solberg, and Erlend Morisbak. Salt diapirs without well defined boundaries - a feasibility study of semi-automatic detection. *Geophysical Prospecting*, 59:682–696, 2011.

[3] David Blacknell and Luc Vignaud. Atr of ground targets: Fundametals and key challenges. In *RADAR Automatic Target Recognition (ATR) and Non-Cooperative Target Recognition (NCTR)*, pages 1.1–1.34, 2012. RTO-EN-SET-172-2012.

[4] J. Engström, M. Trieb, and G. A. Shippey. Tools for classification of mine-like objects in synthetic aperture sonar images. In *Proceedings of the Institude of Acoustics*, volume 29, pages 259–269, 2007.

[5] R. Fandos and A. M. Zoubir. Optimal feature set for automatic detection and classification of underwater objects in sas images. *IEEE Journal on Selected Topics in Signal Processing*, 5(3):454–468, 2011.

[6] J. Fawcett and M. et al Couillard. Computer-aided detection and classification of sidescan sonar images from te citadel trial. In *Proceedings of the Institute of Acoustics*, volume 29, pages 3–10, 2007.

[7] Roy Edgar Hansen. Introduction to sonar. http://www.uio.no/studier/emner/matnat/ifi/INF-GEO4310/h11/undervisningsmateriale/sonar_introduction_2011_compressed.pdf. Course material to INF-GEO4310, University of Oslo, Autumn 2011.

[8] M. S. Horritt. A statistical active contour model for sar image segmentation. *Image and Vision Computing*, 17(3-4):213–224, 1999.

[9] Shawn Lankton. Region based active contour segmentation. http://www.shawnlankton.com. Accessed: 11/01/2013.

[10] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Sonar image segmentation using an unsupervised hierarchical mrf model. *IEEE Trans. Image Process.*, 9(7):1216–1231, 2000.

[11] Vincent L. Myers. Image segmentation using iteration and fuzzy logic. In *Proceedings of CAD/CAC Conference, Halifax, N.S.*, 2007.

[12] Chris Oliver and Shaun Quegan. *Understanding Synthetic Aperture Radar Images*. SciTech Publishing Inc., Raleigh, NC, 2004.

[13] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[14] L. Rabiner and R. Schafer. *Digital processing of speech signals*. Prentice Hall, New Jersey, 1978.

[15] S. Reed, Y. Petillot, and J. Bell. An automatic approach to the detection and extraction of mine features in sidescan sonar. *IEEE J. Ocean. Eng.*, 28(1):90–105, 2003.

[16] Alireza Vard, Kamal Jamshidi, and Naser Movahhedinia. Small object detection in cluttered image using a correlation based active contour model. *Pattern Recognition Letters*, 33:543–553, 2012.

[17] Wikipedia. Definition of a oil drum container. http://en.wikipedia.org/wiki/Drum_(container). Accessed: 02/02/2013.

[18] G. Winkler and V. Liebscher. Smoothers for discontinouos signals. *Journal of Nonparametric Statistics*, 14:203–222, 2002.